



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**EXPERIMENTAL AND THEORETICAL PERFORMANCE
OF A PARTICLE VELOCITY VECTOR SENSOR IN A
HYBRID ACOUSTIC BEAMFORMER**

by

Jeffrey V. Caulk

December 2009

Thesis Advisor:
Co-Advisor:

Daphne Kapolka
Kevin B. Smith

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE December 2009	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE Experimental and Theoretical Performance of a Particle Velocity Vector Sensor in a Hybrid Acoustic Beamformer			5. FUNDING NUMBERS	
6. AUTHOR(S) Jeffrey V. Caulk, CDR, USN				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT <p>Acoustic measurements have traditionally relied exclusively on sound pressure sensors. This research investigated the performance of Microflown 3D hybrid pressure and acoustic particle velocity sensors in a linear array. Each Microflown sensor has three output channels proportional to the acoustic particle velocity in the three, nominally orthogonal, directions in addition to an output from an omnidirectional pressure microphone. The linear array was formed with a conventional omnidirectional microphone as the center element and two Microflown sensors located 17.2cm away on either side. The Microflown acoustic particle velocity channels were characterized first by the amplitude and phase relationship of their transfer functions relative to their co-located pressure microphone. The transfer function between the Microflown pressure hydrophones and the conventional center microphone was also measured. This enabled the amplitude and phase of all channels to be expressed relative to the center microphone signal. Beamforming was carried out in the frequency domain by applying the appropriate weight and phase delay for the desired steer angle. The bandwidth of the beamformer was limited from about 300Hz to 1.5kHz. At lower frequencies, insufficient signal to noise limited the coherence required to establish the transfer functions while at higher frequencies the phase of the particle velocity transfer functions grew increasingly sensitive to orientation angle. Experiments carried out in the Naval Postgraduate School Anechoic Chamber using single and multiple acoustic sources compared extremely well to the theoretical performance. The addition of hybrid pressure and particle velocity sensors proved successful in eliminating the bearing ambiguity inherent in a linear array of omnidirectional sensors with no change in orientation, no complicated post-processing and no additional time expended.</p>				
14. SUBJECT TERMS particle velocity, beamforming, beam pattern, acoustic array, acoustic vector sensor, Microflown, hybrid array, frequency spectrum, linear array, directional			15. NUMBER OF PAGES 173	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**EXPERIMENTAL AND THEORETICAL PERFORMANCE OF A PARTICLE
VELOCITY VECTOR SENSOR IN A HYBRID ACOUSTIC BEAMFORMER**

Jeffrey V. Caulk
Commander, United States Navy
B.A., University of California at Los Angeles, 1990

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING ACOUSTICS

from the

**NAVAL POSTGRADUATE SCHOOL
December 2009**

Author: Jeffrey V. Caulk

Approved by: Daphne Kapolka
Thesis Advisor

Kevin B. Smith
Co-Advisor

Daphne Kapolka
Chair, Engineering Acoustics Academic Committee

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Acoustic measurements have traditionally relied exclusively on sound pressure sensors. This research investigated the performance of Microflown 3D hybrid pressure and acoustic particle velocity sensors in a linear array. Each Microflown sensor has three output channels proportional to the acoustic particle velocity in the three, nominally orthogonal, directions in addition to an output from an omnidirectional pressure microphone. The linear array was formed with a conventional omnidirectional microphone as the center element and two Microflown sensors located 17.2cm away on either side. The Microflown acoustic particle velocity channels were characterized first by the amplitude and phase relationship of their transfer functions relative to their co-located pressure microphone. The transfer function between the Microflown pressure hydrophones and the conventional center microphone was also measured. This enabled the amplitude and phase of all channels to be expressed relative to the center microphone signal. Beamforming was carried out in the frequency domain by applying the appropriate weight and phase delay for the desired steer angle. The bandwidth of the beamformer was limited from about 300Hz to 1.5kHz. At lower frequencies, insufficient signal to noise limited the coherence required to establish the transfer functions while at higher frequencies the phase of the particle velocity transfer functions grew increasingly sensitive to orientation angle. Experiments carried out in the Naval Postgraduate School Anechoic Chamber using single and multiple acoustic sources compared extremely well to the theoretical performance. The addition of hybrid pressure and particle velocity sensors proved successful in eliminating the bearing ambiguity inherent in a linear array of omnidirectional sensors with no change in orientation, no complicated post-processing and no additional time expended.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
II.	THEORY	3
A.	BEAMFORMING VELOCITY SENSORS.....	3
B.	USE OF TRANSFER FUNCTIONS FOR BEAMFORMING.....	7
C.	CHARACTERIZING THE SENSORS.....	10
III.	THE SENSORS AND TESTING EQUIPMENT	13
A.	HARDWARE AND SOFTWARE	13
B.	THE APPARATUS	14
C.	ARRAY COORDINATE SYSTEM.....	18
IV.	TESTING METHODOLOGY AND INITIAL TROUBLESHOOTING.....	21
A.	SAMPLING FREQUENCY	21
B.	VOLTAGE INDICATIONS OF ROTATOR POSITION	22
C.	NOISE AND GROUNDING	26
V.	BEAM PATTERNS OF THE MICROFLOWN SENSORS.....	29
A.	METHOD.....	29
B.	POST-PROCESSING TO DETERMINE POLAR BEAM PATTERNS.....	30
C.	RESPONSE AND BEAM PATTERNS OF THE INDIVIDUAL VELOCITY ELEMENTS.....	32
D.	ORTHOGONALITY OF THE VELOCITY ELEMENTS.....	33
VI.	CHARACTERIZING THE RESPONSE OF THE VELOCITY ELEMENTS AND CALCULATING TRANSFER FUNCTIONS	35
A.	METHOD.....	35
B.	CALCULATING THE TRANSFER FUNCTIONS	38
C.	COHERENCE	49
D.	AMPLITUDE	49
E.	PHASE	51
F.	RESPONSE TO VARIOUS SIGNAL AMPLITUDES.....	51
G.	RESPONSE TO SIGNALS ARRIVING FROM DIFFERENT ASPECTS	55
H.	DIFFERENCE BETWEEN PRESSURE ELEMENT TRANSFER FUNCTIONS OF THE RED AND GREEN MRA ORIENTATIONS	58
VII.	BEAMFORMER	61
A.	METHOD.....	61
B.	RUNNING THE BEAMFORMER.....	62
C.	STATIC TEST WITH SINGLE SOURCE.....	68
D.	DYNAMIC TEST WITH SINGLE SOURCE.....	71
E.	DIRECTIONALITY / BEARING AMBIGUITY	73
F.	BEARING ACCURACY	74
G.	ALTERNATE FREQUENCY PERFORMANCE	76

H.	MULTIPLE SOURCE DISCRIMINATION	79
I.	LIVE TARGET.....	81
VIII.	CONCLUSION	87
APPENDIX A.	TECHNICAL SPECIFICATIONS	89
A.	MICROFLOWN ULTIMATE SOUND PROBE.....	89
B.	MICROFLOWN SIGNAL CONDITIONER.....	91
C.	ACO PACIFIC MICROPHONE	96
1.	Cartridge Microphone Model 7046	96
2.	Preamplifier ½-inch Model 4012	97
D.	NATIONAL INSTRUMENTS CHASSIS	98
E.	NATIONAL INSTRUMENTS DATA ACQUISITION MODULES.....	100
F.	VIFA LOUDSPEAKER DRIVER	105
G.	CALCULATED INDUSTRIES LASER DISTANCE METER	107
H.	ARRAY SENSOR FIXTURE MACHINING PARAMETERS.....	108
APPENDIX B.	COMPUTER CODE	111
A.	LABVIEW VI	111
B.	MATLAB – POLAR_323.M.....	113
C.	MATLAB –TRANSFER_FUNCTION_CAL_323.M	120
D.	MATLAB –TRANSFER_FUNCTION_CAL_323_GRN_MULTI.M ...	126
E.	MATLAB –TRANSFER_FUNCTION_CAL_323_PRS_MULTI.M	131
F.	MATLAB – AVERAGE_TRANSFER_FUNCTIONS.M	135
G.	MATLAB – MICROFLOWNARRAY_12CHAN_BEAMFORMER.M	141
LIST OF REFERENCES.....		151
INITIAL DISTRIBUTION LIST		155

LIST OF FIGURES

Figure 1.	Hybrid Array Construction and Coordinate System Showing Relative Sensor Positions.....	3
Figure 2.	Photograph of Array Monitoring Station Outside of Anechoic Chamber.....	13
Figure 3.	Sketch of the Array Fixture	15
Figure 4.	The Array Fixture with Sensors and Laser Distance Meter installed ..	16
Figure 5.	The Microflown Sensor Elements.....	17
Figure 6.	Photograph of the Array Fixture with Axes Labeled	18
Figure 7.	Array Orientations and Axes of Rotation for Testing, Aco Microphone at Center with Microflown Sensors On Either End	19
Figure 8.	Method for Determining Polar Angles, View From x Axis.....	20
Figure 9.	Measuring the Polar Angles With Laser Distance Meters.....	20
Figure 10.	Photograph of the Rotator and Rotational Position Indicator Controls	23
Figure 11.	Rotational Position Indicator Battery.....	23
Figure 12.	Complete Data Trace of the Voltage Position Signal.....	24
Figure 13.	Close Up of the Voltage Position Signal with Stepped Transitions.....	25
Figure 14.	Noise and Overshoot of Voltage Position Signal	26
Figure 15.	Sensor 323 Low Frequency Noise While Rotating	27
Figure 16.	Sensor 324 Low Frequency Noise While Rotating	28
Figure 17.	Source Signal Used for Polar Plot Testing	29
Figure 18.	Uncorrected Polar Plots.....	31
Figure 19.	Realigned Polar Plots	31
Figure 20.	Realigned Normalized Polar Plots	32
Figure 21.	Ensuring the Array is Coplanar with Source, Note Foam Cover	36
Figure 22.	Aligning The Sensors in the Array Using an Oscilloscope.....	36
Figure 23.	Source Signal Used for Determining Transfer Functions.....	37
Figure 24.	Frequency Components By Element With Array Aligned to Blue MRA	37
Figure 25.	Raw Pressure Element Transfer Functions for Sensor 323.....	40
Figure 26.	Raw Pressure Element Transfer Functions for Sensor 324.....	40
Figure 27.	Raw Blue Velocity Element Transfer Functions for Sensor 323	41
Figure 28.	Raw Blue Velocity Element Transfer Functions for Sensor 324	41
Figure 29.	Raw Green Velocity Element Transfer Functions for Sensor 323	42
Figure 30.	Raw Green Velocity Element Transfer Functions for Sensor 324	42
Figure 31.	Raw Red Velocity Element Transfer Functions for Sensor 323.....	43
Figure 32.	Raw Red Velocity Element Transfer Functions for Sensor 324.....	43
Figure 33.	Final Averaged Pressure Element Transfer Function for Sensor 323.	45
Figure 34.	Final Averaged Pressure Element Transfer Function for Sensor 324.	45
Figure 35.	Final Averaged Blue Velocity Element Transfer Function for Sensor 323	46

Figure 36.	Final Averaged Blue Velocity Element Transfer Function for Sensor 324	46
Figure 37.	Final Averaged Green Velocity Element Transfer Function for Sensor 323	47
Figure 38.	Final Averaged Green Velocity Element Transfer Function for Sensor 324	47
Figure 39.	Final Averaged Red Velocity Element Transfer Function for Sensor 323	48
Figure 40.	Final Averaged Red Velocity Element Transfer Function for Sensor 324	48
Figure 41.	Raw and Corrected Signal Energies of a 1kHz Signal, Sensor 324 Blue Element	49
Figure 42.	Raw and Corrected Signal Energies of a 1kHz Signal, Sensor 324 Green Element	50
Figure 43.	Raw and Corrected Signal Energies of a 1kHz Signal, Sensor 324 Red Element.....	50
Figure 44.	Sensor 324 Blue Element Transfer Function at Various Signal Amplitudes.....	52
Figure 45.	Sensor 324 Red Element Transfer Function at Various Signal Amplitudes.....	52
Figure 46.	Sensor 324 Blue Element Transfer Function at Various Signal Amplitudes, Ignoring 1 Volt Signal.....	53
Figure 47.	Sensor 324 Blue Element Transfer Function at Various Signal Amplitudes in Band of Interest, Ignoring 1 Volt Signal.....	53
Figure 48.	Sensor 324 Red Element Transfer Function at Various Signal Amplitudes, Ignoring 1 Volt Signal.....	54
Figure 49.	Sensor 324 Red Element Transfer Function at Various Signal Amplitudes in Band of Interest, Ignoring 1 Volt Signal.....	54
Figure 50.	Sensor 324 Blue Element Transfer Function at Various Aspects	55
Figure 51.	Sensor 324 Blue Element Transfer Function at Various Aspects in Band of Interest	56
Figure 52.	Sensor 324 Blue Element Transfer Function at Various Aspects Centered at 1kHz.....	56
Figure 53.	Sensor 324 Red Element Transfer Function at Various Aspects.....	57
Figure 54.	Sensor 324 Red Element Transfer Function at Various Aspects in Band of Interest	57
Figure 55.	Sensor 324 Red Element Transfer Function at Various Aspects Centered at 1kHz.....	58
Figure 56.	Sensor 323 Pressure Element Transfer Function at Green and Red Element MRAs	59
Figure 57.	Sensor 324 Pressure Element Transfer Function at Green and Red Element MRAs	60
Figure 58.	Theoretical 3D Beampattern For an N Element Omnidirectional Linear Array	65

Figure 59.	Normalized 3D Beampatterns of the Omni Array With a 1kHz Source At $\theta = 90^\circ$ and $\phi = 0^\circ$ Showing Bearing Ambiguity.....	65
Figure 60.	Normalized 3D Beampatterns of the Hybrid Array With a 1kHz Source At $\theta = 90^\circ$ and $\phi = 0^\circ$ Showing Beam Focusing.....	66
Figure 61.	Beampattern of the Omni and Hybrid Arrays With a Source of 1kHz At $\theta = 90^\circ$ and $\phi = 0^\circ$ on the Same Intensity Scale	67
Figure 62.	3dB Beamwidth Beampattern of the Omni and Hybrid Arrays With a Source of 1kHz at $\theta = 90^\circ$ and $\phi = 0^\circ$ on Respective Intensity Scales.....	67
Figure 63.	Comparison of Hybrid Array Beam Patterns for a 1kHz Signal Placed at Various Relative Aspects in θ Along the ϕ Planar Horizon	69
Figure 64.	Comparison of Omni and Hybrid Array Beam Patterns for a 1kHz Signal Placed at The Three Primary Velocity Element MRAs.	70
Figure 65.	Source Tracking of Omni Array vs. Hybrid Array Using a 1kHz Source Moving About the x Axis. (Arrow Denotes Direction of Travel)	72
Figure 66.	Source Tracking of Hybrid Array Showing Rapid Bearing Resolution Over the Pole (Elapsed Time ~ 10 sec).....	73
Figure 67.	3dB Beamwidth Differences of Both Arrays from a 1kHz Source Plotted with Color Scales Relative to Each Array	74
Figure 68.	3dB Beampattern of The Hybrid Array for an Overhead 1kHz Signal.	75
Figure 69.	Bearing Accuracy of The Hybrid Array for a 1kHz Signal, Using Half-Beamwidth Method	76
Figure 70.	Beampatterns of the Arrays with a 1.5kHz Source	77
Figure 71.	Beampattern with a 1.5kHz Source Showing Sidelobes.....	77
Figure 72.	Beampatterns of the Arrays with a 500Hz Source.....	78
Figure 73.	Beampatterns of the Arrays with a 300Hz Source.....	78
Figure 74.	Sidelobes and Beamwidth Broadening with Various Array Length/Frequency Ratios of Continuous Line Arrays (From: Ziomek, 1994)	78
Figure 75.	Beampatterns of the Hybrid Array with Two Sources at the Same Frequency and a Spatial Separation of 60° in θ	80
Figure 76.	Beampatterns of the Hybrid Array with Two Sources at Different Frequencies and a Spatial Separation of 60° in θ Processed with Separate Center Frequencies	80
Figure 77.	Beampatterns of the Hybrid Array with Two Sources at Different Frequencies and a Spatial Separation of 20° in ϕ Processed with Separate Center Frequencies	81
Figure 78.	Frequency Content of the Live Signal - Turboprop Target.....	83
Figure 79.	Frequency Content of the Live Signal - Jet Target	83
Figure 80.	LOFARGrams of Turboprop Target for ACO Microphone and Sensor 323 Green Velocity Element.....	84

Figure 81.	Low Frequency Region of Sample Pressure and Velocity Element Transfer Functions.....	85
Figure 82.	Erratic Response of Successive Trials in Low Frequency Region of Sample Pressure and Velocity Element Transfer Functions.....	86

LIST OF TABLES

Table 1.	Binary File Sensor Element Channel Assignments	22
Table 2.	Angular Alignments by Element Used in Determining Transfer Functions.....	38
Table 3.	Final Transfer Function Calculations by Element	44

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

First, I would like to thank my lovely wife Mia for her years of steadfast support, not only here at NPS, but throughout my Naval career. Her strength and conviction set positive examples for me in all endeavors.

Thank you to Hans de Bree at Microflown Holland for allowing me the use of his wonderful sensors. I hope you find something useful here.

Thank you to my thesis advisors Daphne Kapolka and Kevin Smith. Your advice and counsel kept me on track and, at times, sane. You helped me troubleshoot when I was too close to the problem to see the cause.

Thank you to my professors, who made all this crazy physics stuff interesting and, at times, enjoyable. Thanks also to my classmates, who welcomed in an old man and assisted me in shaking off the math cobwebs from an 18-year-old Bachelor of Arts.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Acoustic measurements traditionally relied on sound pressure sensors. This research investigated the performance of the Microflow 3D hybrid pressure and acoustic particle velocity sensors in a linear array. These are vector sensors that provide a directional output for each of the three orthogonal particle velocity channels in addition to an omnidirectional pressure output. These sensors are designed for use in air, but the general principles should translate to sensors designed for use in water. Characterization was performed and transfer functions were determined in the Naval Postgraduate School Anechoic Chamber. The characterization consisted of three parts: 1) determining the individual element beam patterns of the four separate elements 2) determining the effect changing signal amplitude and aspect had on the responses of each sensor element and 3) computing transfer functions for each sensor element referenced to a standard calibrated pressure microphone. The Microflow sensors were incorporated into a hybrid three-element array with a calibrated traditional pressure sensor to determine array beamforming performance using single and multiple acoustic sources. The beamforming routine used linear summation with directional weighting of the velocity elements to their respective axes of maximum response.

A traditional linear array using only omnidirectional sensors cannot resolve bearing to the source in three dimensions upon initial detection. In general practice, it requires changing the orientation of the array or obtaining a cross fix from a second search platform in order to resolve the bearing. In the hybrid array, with the vector sensors adding direction components in three dimensions, bearing ambiguity is immediately resolved with no change in orientation, no complicated post-processing and no additional time expended. The hybrid array resolved bearing to the source well. It was also able to track moving targets. The linear summing method used showed good ability to discriminate multiple targets when they were at different frequencies.

THIS PAGE INTENTIONALLY LEFT BLANK

II. THEORY

A. BEAMFORMING VELOCITY SENSORS

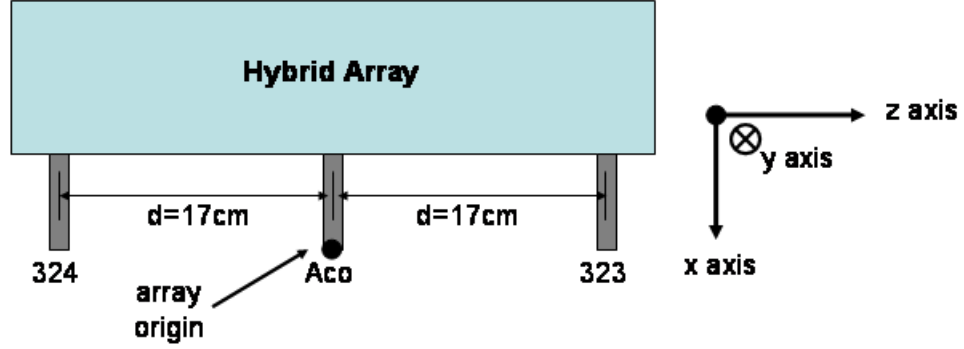


Figure 1. Hybrid Array Construction and Coordinate System Showing Relative Sensor Positions

Figure 1 shows the array as constructed, with the inter-element spacing, d , optimized for a design frequency of 1kHz. Array axes and origin are also delineated. The beamformer algorithm was implemented in the frequency domain with a basic linear sum of the complex weighted channels. The only difference required for accommodating the particle velocity channels is to apply an additional weighting to take advantage of the directionality of the velocity elements. Let sensor 1 be the Microflown sensor 323, sensor 2 be the Aco pressure microphone in the center, and sensor 3 be the Microflown sensor 324.

The elements are designated as follows:

Sensor 1:

e_{11} is the pressure output

e_{12} is the blue velocity sensor output

e_{13} is the green velocity sensor output

e_{14} is the red velocity sensor output.

Sensor 2:

e_{21} is the pressure microphone output

e_{22} through e_{24} are zero

(since the Aco microphone has no velocity outputs)

Sensor 3:

e_{31} is the pressure output

e_{32} is the blue velocity sensor output

e_{33} is the green velocity sensor output

e_{34} is the red velocity sensor output

The direction axes of the Microflow sensors are not perfectly orthogonal nor exactly aligned with the x , y , z axes as defined by the holding fixture. In practice, this had little effect and was confirmed to be negligible through testing. In this discussion, the elements are treated as perfectly aligned and orthogonal.

Let (θ_{12}, ϕ_{12}) be the direction of the Maximum Response Axis (MRA) for the blue vector sensor on Sensor 1, (θ_{13}, ϕ_{13}) be the direction of the MRA for the green vector sensor on Sensor 1, (θ_{14}, ϕ_{14}) be the direction of the MRA for the red vector sensor on Sensor 1, etc., The velocity element unit vectors are given in terms of the direction cosines by

$$\hat{u}_{e12} = \sin \theta_{12} \cos \phi_{12} \hat{x} + \sin \theta_{12} \sin \phi_{12} \hat{y} + \cos \theta_{12} \hat{z} \quad (\text{Eq. 2.1})$$

$$\hat{u}_{e13} = \sin \theta_{13} \cos \phi_{13} \hat{x} + \sin \theta_{13} \sin \phi_{13} \hat{y} + \cos \theta_{13} \hat{z} \quad (\text{Eq. 2.2})$$

$$\hat{u}_{e14} = \sin \theta_{14} \cos \phi_{14} \hat{x} + \sin \theta_{14} \sin \phi_{14} \hat{y} + \cos \theta_{14} \hat{z} \quad (\text{Eq. 2.3})$$

$$\hat{u}_{e32} = \sin \theta_{32} \cos \phi_{32} \hat{x} + \sin \theta_{32} \sin \phi_{32} \hat{y} + \cos \theta_{32} \hat{z} \quad (\text{Eq. 2.4})$$

$$\hat{u}_{e33} = \sin \theta_{33} \cos \phi_{33} \hat{x} + \sin \theta_{33} \sin \phi_{33} \hat{y} + \cos \theta_{33} \hat{z} \quad (\text{Eq. 2.5})$$

$$\hat{u}_{e34} = \sin \theta_{34} \cos \phi_{34} \hat{x} + \sin \theta_{34} \sin \phi_{34} \hat{y} + \cos \theta_{34} \hat{z} . \quad (\text{Eq. 2.6})$$

The beamformer combines the nine element channels with electrical outputs, with the other three channels, e_{22} through e_{24} , being dummies.

There are three steps to beamform the array in a steer direction (θ_s, ϕ_s) . First, the amplitude and phase of each of the eight Microflown vector sensor channels relative to the Aco microphone, i.e., the transfer functions, need to be established. The transfer functions between particle velocity sensors and pressure sensors are measured along the MRA of the velocity sensor. It is necessary to determine whether or not the transfer functions are functions of the signal amplitude and whether or not the phase of the vector sensor transfer functions relative to the pressure sensor is a function of the angle of incidence of the sound wave. A 180° phase shift in the transfer function is expected when sound is coming from the $-MRA$ direction, but any possible off axis phase dependence needs to be determined.

For the sake of the beamforming theory, we assume here that the transfer functions are not dependent on source amplitude and that the phase is independent of both polar and azimuthal angle. The restrictions on the validity of these assumptions will be discussed in Chapter VI. The signals from each velocity and pressure element on the Microflown sensors are translated to the proper amplitude and phase for beamforming by using the measured transfer function between it and the center ACO microphone. This was done in two steps by measuring the transfer function between each of the velocity sensors relative to the co-located pressure sensors and then using the transfer function between the Microflown pressure sensor and the ACO. For example, the overall transfer function between the blue 323 sensor and the ACO is given by

$$\hat{H}_{12}(k) = \frac{\hat{X}_{21}(k)}{\hat{X}_{12}(k)}. \quad (\text{Eq. 2.7})$$

This was computed from the measured transfer functions as,

$$\hat{H}_{12}(k) = \frac{\hat{X}_{21}(k) \hat{X}_{11}(k)}{\hat{X}_{11}(k) \hat{X}_{12}(k)}. \quad (\text{Eq. 2.8})$$

Using the overall transfer function, the amplitude and phase of the blue velocity signal on sensor 1 can be converted in the frequency domain as

$$\hat{X}_{c12}(k) = \hat{X}_{12}(k) \hat{H}_{12}(k), \quad (\text{Eq. 2.9})$$

where $\hat{X}_{c12}(k)$ is the corrected version of e_{12} in the frequency domain.

The next step is to adjust the weight of the vector sensors to steer the beampattern in the desired direction. The unit vector in the steer direction is given by

$$\hat{u}_s = \sin \theta_s \cos \phi_s \hat{x} + \sin \theta_s \sin \phi_s \hat{y} + \cos \theta_s \hat{z}. \quad (\text{Eq. 2.10})$$

The amplitude weighting of e_{12} should be

$$\hat{u}_s \cdot \hat{u}_{e12}. \quad (\text{Eq. 2.11})$$

This also accounts for the 180° phase shift from +MRA to -MRA.

The final step is the conventional beamforming sum, i.e., taking the phase shift due to the spatial position of the sensors into account. Given a distance of d between sensors, the phase shift between the Microflown sensors and the Aco pressure sensor is given by

$$\alpha = \exp \frac{i 2 \pi (m-2) k f_s d \cos \theta_s}{N c}, \quad (\text{Eq. 2.12})$$

where m is the sensor number, k is the frequency bin number, f_s is the sampling frequency, N is the Digital Fourier Transform length, c is the speed of sound, and θ_s is the polar angle.

The output of the beamformer steered in direction \hat{u}_s , is given by

$$\hat{S}(k) = \sum_{m=1}^3 \sum_{n=1}^4 \hat{X}_{c mn}(k) (\hat{u}_{emn} \cdot \hat{u}_s) \exp \frac{-i 2 \pi (m-2) k f_s d \cos \theta_s}{N c}. \quad (\text{Eq. 2.13})$$

All the pressure microphones are nominally omnidirectional, therefore

$$\hat{u}_{e11} \cdot \hat{u}_s = \hat{u}_{e21} \cdot \hat{u}_s = \hat{u}_{e31} \cdot \hat{u}_s = 1. \quad (\text{Eq. 2.14})$$

The Aco microphone has no velocity elements, therefore

$$\hat{X}_{22}(k) = \hat{X}_{23}(k) = \hat{X}_{24}(k) = 0 \quad (\text{Eq. 2.15})$$

The Aco is also the reference element, therefore

$$\hat{X}_{c_{21}}(k) = \hat{X}_{21}(k). \quad (\text{Eq. 2.16})$$

B. USE OF TRANSFER FUNCTIONS FOR BEAMFORMING

Theoretically, the acoustic particle velocity of a planar sound wave differs from its pressure by a factor equal to the specific acoustic impedance of the medium. In practice, the amplitude and phase sensitivity of each sensor used to measure these quantities must be known in order to determine the actual pressure and particle velocity. Although detailed calibration curves for the Microflown sensors were available, any influence on the sensitivity of the sensor due to its placement in the jig would not be taken into account with the free-field calibration data. Furthermore, the sensitivity of the conventional Aco microphone also needs to be taken into account before beamforming. To alleviate the need for using the calibration curves for each sensor to correct the time domain output to the actual pressure or particle velocity, the transfer functions were measured between each of the Microflown channels and the Aco microphone. This made it possible to scale all Microflown channels to a signal commensurate with the Aco signal. Both the correction and the subsequent beamforming were carried out in the frequency domain. The discussion which follows explains the theoretical difference between the acoustic particle velocity and the pressure for the general case of a spherical wave and then goes on to explain how the transfer functions were used to scale the Microflown signals prior to beamforming.

To start, the direction of the actual velocity element MRAs are given in conventional direction order x, y, z . The numerical order of the elements is consistent with the discussion above and the array coordinate system below. They are based on the channel assignments on the Microflown signal conditioner. Thus, the coordinate and numerical schemes are not in the same order. The blue sensor is first in numerical order after the pressure sensor. Therefore, it is assigned element number 2, but has its MRA along the z axis. The green element comes next, so it is assigned element number 3, but it has its

MRA along the x axis. Finally, the red sensor is last, so it is assigned element number 4 and has its MRA along the y axis.

Assume the velocity elements are perfectly aligned with the coordinate axes, so

$$\hat{u}_{e13} = \hat{x} \quad (\text{Eq. 2.17})$$

$$(\theta_{13}, \phi_{13}) = (90^\circ, 90^\circ) \quad (\text{Eq. 2.18})$$

$$\hat{u}_{e14} = \hat{y} \quad (\text{Eq. 2.19})$$

$$(\theta_{14}, \phi_{14}) = (90^\circ, 0^\circ) \quad (\text{Eq. 2.20})$$

$$\hat{u}_{e12} = \hat{z} \quad (\text{Eq. 2.21})$$

$$(\theta_{12}, \phi_{12}) = (0^\circ, \text{any}) \quad (\text{Eq. 2.22})$$

then,

$$V_{13} = V \sin \theta \cos \phi \quad (\text{Eq. 2.23})$$

$$V_{14} = V \sin \theta \sin \phi \quad (\text{Eq. 2.24})$$

$$V_{12} = V \cos \theta. \quad (\text{Eq. 2.25})$$

Where $V_{13,14,12}$ are the amplitudes of the velocity responses of sensor 323 on the green, red and blue elements respectively, for an incoming signal in the direction of (θ, ϕ) . The value V is the magnitude of the acoustic particle velocity which would be measured by a velocity sensor whose MRA corresponded to the direction of the incoming wave.

Assume a normalized spherical wave is incident upon the sensors defined by

$$p = A \frac{e^{ikr}}{r} = \frac{A}{r} e^{i\phi_p}, \quad (\text{Eq. 2.26})$$

where A is the amplitude of the signal and kr is the phase of the signal relative to its origin. The corresponding velocity response is

$$\bar{v} = -\frac{i}{\omega\rho} \nabla p = -\frac{iA}{k\rho c} \frac{e^{ikr}}{r^2} (ikr - 1) \hat{r} = \frac{p}{\rho c} \left(1 + \frac{i}{kr} \right) \hat{r} \quad (\text{Eq. 2.27})$$

Neglecting direction, the expression is

$$v = -\frac{iA}{\omega\rho} \frac{e^{ikr}}{r^2} (ikr - 1) = V e^{i\phi_v}, \quad (\text{Eq. 2.28})$$

where

$$V = \frac{A}{\omega \rho r^2} \sqrt{(kr)^2 + 1} \quad (\text{Eq. 2.29})$$

$$\text{and } \phi_v = kr + \tan^{-1}\left(\frac{1}{kr}\right). \quad (\text{Eq. 2.30})$$

In the far-field when $kr \gg 1$, these expressions reduce to

$$V \cong \frac{A}{\rho c r}, \quad \phi_v \cong kr,$$

$$\text{and } v \cong \frac{A}{\rho c} \frac{e^{ikr}}{r} = \frac{p}{\rho c} \quad (\text{Eq. 2.31})$$

which are the plane wave limits expected. Therefore, assuming that a particle velocity sensor is oriented to pick up the outgoing radial component of the acoustic wave, its signal will be in phase with the pressure, and the amplitude will differ by a factor of ρc .

Now the measured amplitude and phase of all channels depends not only on whether pressure or particle velocity is being measured, but also on the sensitivity of the sensor. Therefore, there will be some additional amplitude and phase mismatch that needs to be corrected prior to beamforming. To account for this, the transfer function between each velocity channel and its on board pressure sensor is measured along its MRA. Since the three velocity sensors are co-located with their pressure sensor, errors in phase are minimized. These transfer functions can be expressed as

$$\hat{H}_{13}(k) = \frac{\hat{X}_{11}(k)}{\hat{X}_{13}(k)}, \quad (\text{Eq. 2.35})$$

$$\hat{H}_{14}(k) = \frac{\hat{X}_{11}(k)}{\hat{X}_{14}(k)}, \quad (\text{Eq. 2.36})$$

$$\text{and } \hat{H}_{12}(k) = \frac{\hat{X}_{11}(k)}{\hat{X}_{12}(k)}, \quad (\text{Eq. 2.37})$$

where $\hat{H}_{13}(k)$ is the transfer function as a function of bin frequency (k), \hat{X}_{11} is the auto-spectral density of the 323 pressure sensor, and \hat{X}_{13} is the cross-

spectral density between the 323 pressure sensor and the 323 particle velocity sensor oriented along the positive x-axis.

Once these transfer functions are known, the output from each of the vector sensor channels can be scaled to its on board pressure sensor. Assume that the source is in the direction (θ, ϕ) . Letting $\hat{p}_{11}(k)$ be the output of the 323 pressure sensor in the frequency domain and $\hat{v}_{13}(k)$ be the output of the 323 particle velocity sensor, we get the correctly scaled particle velocity signal as:

$$\hat{v}_{13c} = \hat{v}_{13}(k) \hat{H}_{13}(k) = \hat{p}_{11}(k) \sin \theta \cos \phi, \quad (\text{Eq. 2.38})$$

This result is a velocity component, scaled to match the pressure component, and either in phase with the pressure response or 180° out of phase depending on the source position. The amplitude of this velocity component is less than or equal to the pressure due to the response of the velocity sensor to signals coming from an off-MRA direction. Similar results are obtained for the other particle velocity channels.

$$\hat{v}_{14c} = \hat{v}_{14}(k) \hat{H}_{14}(k) = \hat{p}_{11}(k) \sin \theta \sin \phi \quad (\text{Eq. 2.46})$$

$$\hat{v}_{12c} = \hat{v}_{12}(k) \hat{H}_{12}(k) = \hat{p}_{11}(k) \cos \theta \quad (\text{Eq. 2.47})$$

The same procedure is carried out for Sensor 3 with only a change in indices, i.e., $\hat{H}_{32}(k)$, $\hat{H}_{33}(k)$ and $\hat{H}_{34}(k)$. To further reference these channels to the central Aco microphone, the transfer functions between the Microflown pressure sensors and the Aco microphone are measured as well.

C. CHARACTERIZING THE SENSORS

Based on the reasoning above, the following steps were used to characterize the sensors and establish the transfer functions.

1) The pressure sensor transfer functions between the three pressure sensors were determined by exposing them to the same acoustic pressure and

phase by placing them at the same distance from the source signal in order to avoid any path length differences. This produced the transfer functions $\hat{F}_{12}(k)$ and $\hat{F}_{32}(k)$.

2) The velocity sensor transfer functions relative to the onboard pressure sensor for each Microflown vector sensor were determined with the source signal aligned with the MRA for each velocity element separately. This resulted in six different velocity transfer functions denoted by

$$\hat{H}_{12}(k), \hat{H}_{13}(k), \hat{H}_{14}(k), \hat{H}_{32}(k), \hat{H}_{33}(k) \text{ and } \hat{H}_{34}(k).$$

3) The angular relationships of the velocity sensors to their respective MRAs were recorded and stored as the values of

$$(\theta_{12}, \phi_{12}), (\theta_{13}, \phi_{13}), (\theta_{14}, \phi_{14}), (\theta_{32}, \phi_{32}), (\theta_{33}, \phi_{33}) \text{ and } (\theta_{34}, \phi_{34}).$$

These are used to determine absolute orientation of the sensors in the array. During testing, the velocity elements of the vector sensors were determined to be close enough to orthogonal to treat them as such. See the characterization discussion below for details.

4) The corrected values can then be computed from each channel and the appropriate transfer functions as follows:

$$\hat{X}_{c11}(k) = \hat{F}_{12}(k) \hat{X}_{11}(k) \quad (\text{Eq. 2.52})$$

$$\hat{X}_{c12}(k) = \hat{F}_{12}(k) \hat{H}_{12}(k) \hat{X}_{12}(k) \quad (\text{Eq. 2.53})$$

$$\hat{X}_{c13}(k) = \hat{F}_{12}(k) \hat{H}_{13}(k) \hat{X}_{13}(k) \quad (\text{Eq. 2.54})$$

$$\hat{X}_{c14}(k) = \hat{F}_{12}(k) \hat{H}_{14}(k) \hat{X}_{14}(k) \quad (\text{Eq. 2.55})$$

$$\hat{X}_{c31}(k) = \hat{F}_{32}(k) \hat{X}_{31}(k) \quad (\text{Eq. 2.56})$$

$$\hat{X}_{c32}(k) = \hat{F}_{32}(k) \hat{H}_{32}(k) \hat{X}_{32}(k) \quad (\text{Eq. 2.57})$$

$$\hat{X}_{c33}(k) = \hat{F}_{32}(k) \hat{H}_{33}(k) \hat{X}_{33}(k) \quad (\text{Eq. 2.58})$$

$$\text{and } \hat{X}_{c34}(k) = \hat{F}_{32}(k) \hat{H}_{34}(k) \hat{X}_{34}(k). \quad (\text{Eq. 2.59})$$

5) The steering vector was defined as above as

$$\hat{u}_s = \sin \theta_s \cos \phi_s \hat{x} + \sin \theta_s \sin \phi_s \hat{y} + \cos \theta_s \hat{z}. \quad (\text{Eq. 2.10})$$

6) The beamforming expression remains

$$\hat{S}(k) = \sum_{m=1}^3 \sum_{n=1}^4 X_{c_{mn}}(k) (\hat{u}_{emn} \cdot \hat{u}_s) \exp \frac{-i2\pi(m-2)kf_s d \cos \theta_s}{Nc}. \quad (\text{Eq. 2.12})$$

As before, all of the pressure microphones are nominally omnidirectional, therefore

$$\hat{u}_{e11} \cdot \hat{u}_s = \hat{u}_{e21} \cdot \hat{u}_s = \hat{u}_{e31} \cdot \hat{u}_s = 1. \quad (\text{Eq. 2.13})$$

The Aco microphone has no velocity elements, therefore

$$\hat{X}_{22}(k) = \hat{X}_{23}(k) = \hat{X}_{24}(k) = 0. \quad (\text{Eq. 2.14})$$

The Aco is also the reference element, therefore

$$\hat{X}_{c_{21}}(k) = \hat{X}_{21}(k). \quad (\text{Eq. 2.15})$$

III. THE SENSORS AND TESTING EQUIPMENT

A. HARDWARE AND SOFTWARE

The sensors and data acquisition (DAQ) equipment used in testing and evaluation are listed below. Figure 2 shows the monitoring station outside the Anechoic Chamber, consisting of a digital voltmeter to monitor nominal array polar position, the rotator controls and a laptop computer that collects the data streams via the DAQ. The technical specifications and calibration data are included in Appendix A. All data was collected with the Microflown signal conditioner set to High Gain with the correction off. The wire screen covers on the Microflown sensors were removed.

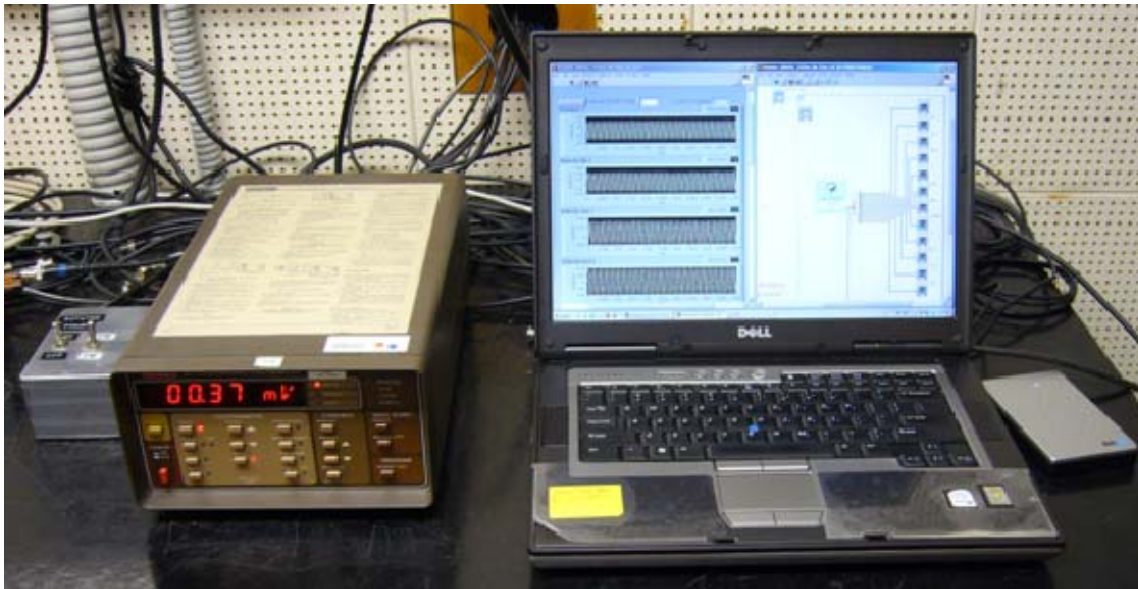


Figure 2. Photograph of Array Monitoring Station Outside of Anechoic Chamber

- Aco Pacific Calibrated Pressure microphone – Cartridge Model 7046
- Aco Pacific 1/2" preamplifier – Model 4012
- Microflown Holland Four Channel Ultimate Sound Probe – Model UT0901-23, designated Sensor 323 below
- Microflown Holland Four Channel Signal Conditioner – Model E0901-23

- Microflown Holland Four Channel Ultimate Sound Probe – Model UT0901-24, designated Sensor 324 below
- Microflown Holland Four Channel Signal Conditioner – Model E0901-24
- National Instruments CompactDAQ USB Chassis – Model NI 9172
- National Instruments Sound and Vibration DAQ Module – Model NI 9234 (3 total)
- Dell D820 Laptop Computer
- Hewlett Packard Function Generator – Model 33120A
- Keithly Programmable Electrometer – Model 617 (Digital Voltmeter)
- Tektronix Digital Oscilloscope – Model TDS 210
- Vifa 5" loudspeaker – model P13WH-00-08, mounted in a sealed 10.6 liter enclosure with no crossover.
- Techron Mono/Stereo Two Channel Amplifier – Model 5507, 240W
- Philips Universal Frequency Counter – Model 6669
- Calculated Industries Laser Distance Meter – Model Prexiso X2 (2 total)
- Johnson Self Leveling Laser Level – Model 40-6620
- Motorized rotator mechanism with gear reduction transmission and potentiometer used as an angular position indicator.
- Data acquisition conducted with National Instruments LabVIEW version 8.5.
- Post-processing and plotting completed using MathWorks MATLAB versions R2007b, R2008b and R2009b.

B. THE APPARATUS

The Microflown sensors have no calibrated registration slots, keyways or even markings on the exterior of the probes to reference the velocity element axes. Therefore, they may only be roughly aligned by eye to a coordinate system. Once aligned and the transfer functions calculated, if they were removed and reinstalled, they would have different orientations relative to the array and to each other. Thus, they would require realignment and calculation of

new transfer functions every time data collection was performed. To eliminate this problem, a fixture was designed to hold the sensors permanently in place.

This fixture needed to secure the sensors against rotation or movement relative to one another, but it also had to be portable and sufficiently small so it did not have a negative affect on the measured sound field due to reflections. After construction, the apparatus had to be capable of being removed and reinstalled in the test chamber or taken to the operational environment without losing positional alignment. Figure 3 shows the design of the fixture (Appendix H contains the measured specifications). Figure 4 shows the completed fixture as constructed.

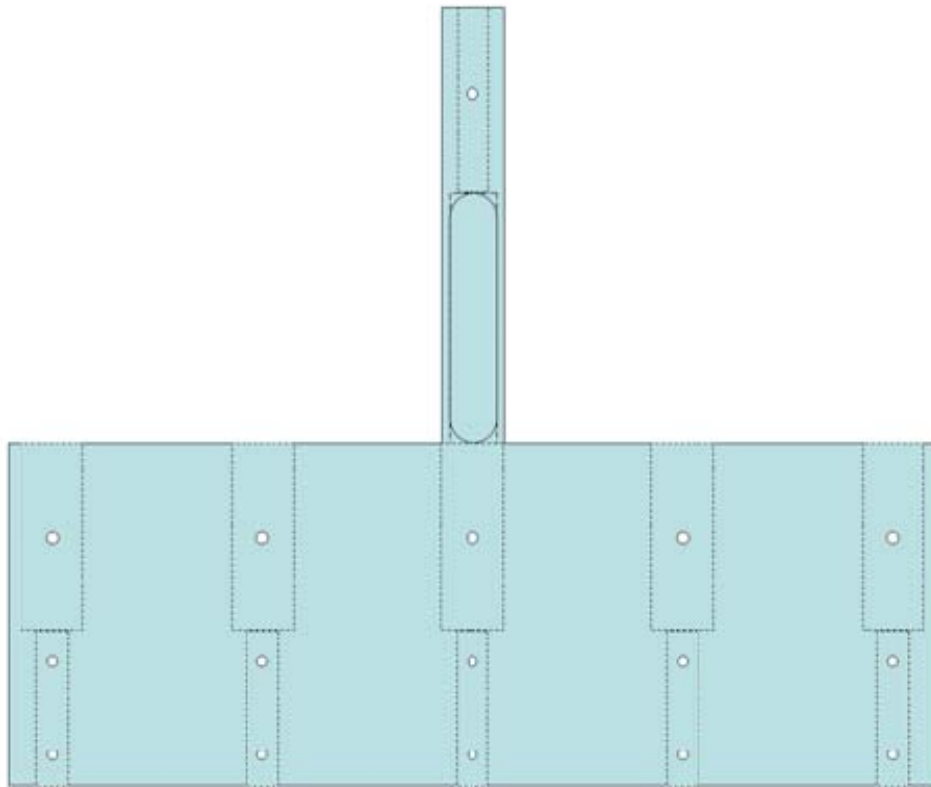


Figure 3. Sketch of the Array Fixture



Figure 4. The Array Fixture with Sensors and Laser Distance Meter installed

The fixture was designed and constructed locally at Naval Postgraduate School. The primary carrier for the three sensors is made from 1 1/2" thick High Density Poly Ethylene (HDPE) plastic. The HDPE was machined with tight tolerances to secure the Microflown sensors in fixed positions well enough to allow removal and reinstallation of the fixture for the various experiments. During initial testing with the sensors mounted in lab clamps, rotational and vibrational noise was evident in the velocity element frequency plots. The mass of the apparatus and the rigid mounting helped to damp out vibrations and quiet this noise a great deal.

The five holes are spaced at 8.5cm. This spacing was chosen for flexibility. With a design frequency of 1kHz, corresponding to a wavelength of 34cm, the half-wavelength spacing for the array elements placed them at 17cm apart. This is the spacing between the outer positions and the center position. The actual distance, measured after construction, from the center of the Aco microphone to the center of the Microflown was 17.2cm. As shown in Figure 5, the velocity elements sit on either side of the 3.5mm central pressure microphone. Therefore, the individual element spacing can vary by about $\pm 2\text{mm}$. At the frequencies tested, this variance is inconsequential.

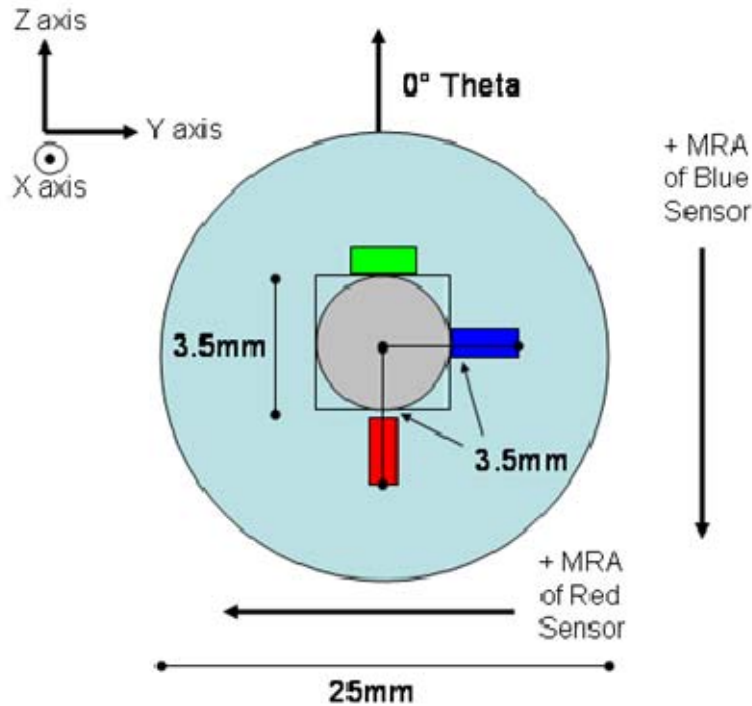


Figure 5. The Microflow Sensor Elements

The holes also needed to pass the sensor cables out the top and allow the cables to be removed if necessary while not disturbing the sensor mountings. Contours and holes were machined around the sensor mounting locations to reduce the mass and to minimize the reflections from flat surfaces on the fixture. After construction and mounting of the sensors, the entire apparatus was covered with 1 1/2" egg crate foam rubber to further reduce reflections.

The central shaft is aluminum tubing. A slot was milled in the side of the tubing to allow the 3/4" plug from the Aco pressure microphone lead to pass out of the shaft and to the DAQ equipment. A steel fitting was press fit into the end of the tubing and secured. A standard laboratory apparatus rod was secured into the steel fitting with a screw. The rod in turn fits into the rotator mechanism in the anechoic chamber for positioning and rotational analysis.

The central shaft may be removed and placed into the upper hole coaxial with any of the sensor mounting holes. This allows the beam patterns of the Microflow sensors, which reside on the outboard stations, to be determined without needing to compensate for off axis rotation and path length differences.

The fixture axes were labeled to avoid confusion. Figure 6 shows the axes of the completed apparatus. When in the rotator the array operated with the x axis pointing down. When capturing live data it was generally in the x up orientation.

C. ARRAY COORDINATE SYSTEM

The array uses standard spherical and Cartesian coordinate systems with the origins co-located at the omnidirectional Aco microphone. The z axis extends along the long axis of the fixture, with positive z projecting from the Aco toward the 323 sensor. The x axis extends along the longitudinal axis of the Aco sensor, with positive x projecting out of the face of the Aco away from the central rotating shaft. The y axis follows standard right-hand convention and projects away from the Aco normal to the large flat side of the fixture. Polar θ and azimuthal ϕ angles are measured in the standard manner, with θ clockwise from positive z toward negative z , and ϕ counter clockwise in the xy plane measured from the x axis. The range of θ is from 0 to π , and ϕ is from $-\pi$ to π .



Figure 6. Photograph of the Array Fixture with Axes Labeled

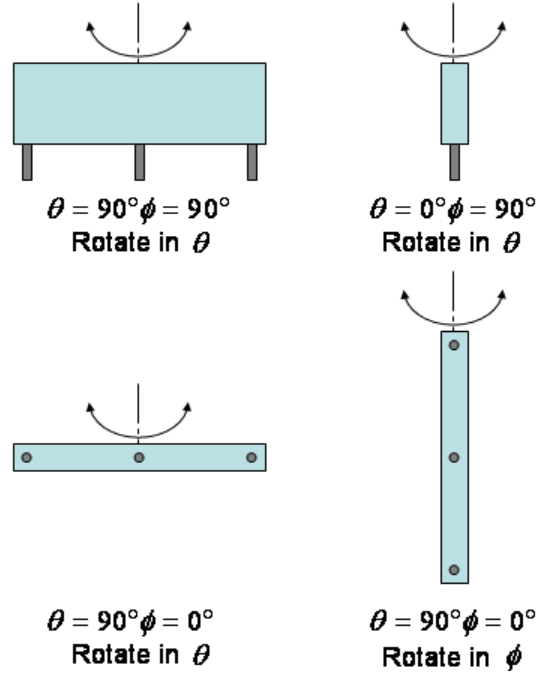


Figure 7. Array Orientations and Axes of Rotation for Testing, Aco Microphone at Center with Microflow Sensors On Either End

Figure 7 shows the various array test orientations with their associated axes of rotation while mounted in the rotator apparatus. A laser ruler was affixed with its base at the origin and aligned with the z axis. Another laser ruler was affixed to the top of the loudspeaker cabinet with its face even with the face of the loudspeaker driver. In anechoic chamber testing, the source and array origin remained at a fixed distance. The array was rotated about the origin and the polar angle determined. To determine the polar angle, the laser on the array was illuminated and a target was placed in front of the 0° point of the z axis where it was marked by the laser. The laser affixed to the loudspeaker was rotated, keeping its face even with the face of the driver, until the spot of the laser was coincident on the target with the spot from the array laser. The two measurements were recorded. The distance from the loudspeaker to the array origin was verified and the polar angle determined using the law of cosines. Due to difficulty in apparatus construction, only azimuthal angles of 0° and 90° were used during testing. Figures 8 and 9 show the method and execution of determining polar angles.

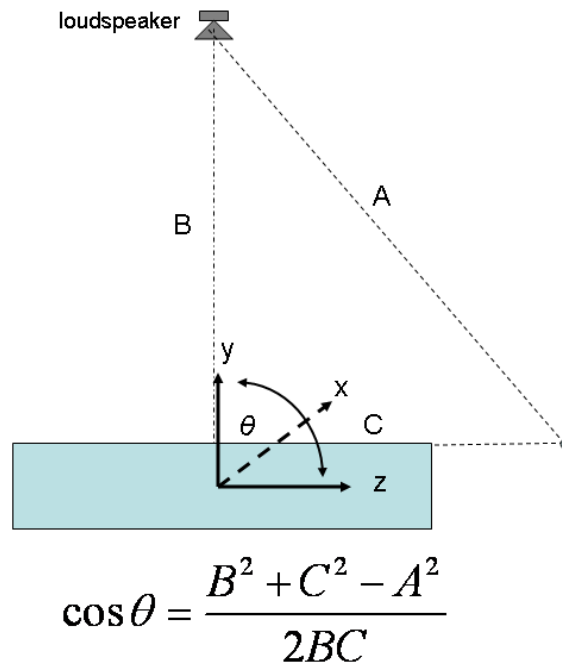


Figure 8. Method for Determining Polar Angles, View From x Axis



Figure 9. Measuring the Polar Angles With Laser Distance Meters

IV. TESTING METHODOLOGY AND INITIAL TROUBLESHOOTING

A. SAMPLING FREQUENCY

The National Instruments Data Acquisition (DAQ) equipment uses a master internal timebase of 13.1072MHz. The sampling rate is controlled by the first installed module with a maximum of 51.2kHz. All other sampling rates must be integer divisions of the maximum according to the formula

$$f_s = \frac{f_M / 256}{n}, \quad (\text{Eq. 4.1})$$

where f_M is the master clock rate and n is an integer from 1 to 31 (National Instruments, n.d).

It is possible to enter non-allowed sampling rate values into the LabVIEW vi window when programming. If an invalid rate is entered, the equipment will adjust the sampling rate to the nearest allowed rate. There is no warning when this is done and the sampling rate is still displayed with the invalid value in the vi window. Care must be taken to use only allowed rates and ensure all software post-processing is done with the actual sampling rates. Curiously, the default rate in LabVIEW is 25kHz, an invalid rate. Processing data files and conducting Fourier transforms using mismatched frequencies created a great deal of confusion early in testing.

The sampling rate chosen for all testing and data collection was 25.6kHz, which produced a Nyquist maximum usable frequency of 12.8kHz, well above the region of interest. This high sampling rate was chosen to ensure that the transfer function was calculated at a sufficiently fine resolution in the frequency domain. At 25.6kHz, a Fourier transform length of 65,536 (2^{16}) corresponding to 2.56s worth of data produced a frequency resolution of 0.391Hz per bin. The data was recorded as twelve simultaneous channels and saved in binary format. Table 1 shows the channel scheme that was used.

DAQ and MATLAB Data Channel Assignments							
Channel	DAQ Function	MATLAB Desig	Coupling	Channel	DAQ Function	MATLAB Desig	Coupling
1	USP 323 Pressure	prs1	AC	7	Second Source	src2	AC
2	USP 323 Blue	blu1	AC	8	Position Voltage	psit	DC
3	USP 323 Green	grn1	AC	9	USP 324 Pressure	prs3	AC
4	USP 323 Red	red1	AC	10	USP 324 Blue	blu3	AC
5	Aco Pressure	prs2	AC	11	USP 324 Green	grn3	AC
6	First Source	src1	AC	12	USP 324 Red	red3	AC

Table 1. Binary File Sensor Element Channel Assignments

The data streams for the polar plots were captured as 12 chunks of 200,000 data points per chunk per channel, for a total of 2,400,000 data points per channel. At the sampling frequency of 25.6kHz, this produced a data stream 93.75 seconds long. The source was a 1kHz tone at 1Vpp.

The data streams for the transfer functions were captured as 40 chunks of 65,536 (2^{16}) data points per chunk per channel, for a total of 2,621,440 data points per channel. At the sampling frequency of 25.6kHz, this produced a data stream 102.4 seconds long. The source was white noise at 5Vpp.

B. VOLTAGE INDICATIONS OF ROTATOR POSITION

The rotating mechanism consisted of an electric motor, gear reduction transmission and a potentiometer. A nominal 1.5V dry cell battery was attached across the contacts of the potentiometer. As the shaft, and thus the potentiometer, rotated the wiper arm moved along the windings, changing the proportional voltage divider created by the potentiometer leads. Voltage readings were taken between the wiper and one contact of the potentiometer. This produced a general estimate of rotational position vs. voltage with a scale of ~1.5V per 360°, or 240mV/degree. The shaft of the mechanism was notched for a keyed screw to secure the shaft at the attachment point. With the sensors installed on the shaft, the voltage transition point from 0V to 1.5V was roughly

designated as $\theta = 0^\circ$. Figures 10 and 11 show the rotator controls and the simplified electrical circuit for monitoring rotation position within.



Figure 10. Photograph of the Rotator and Rotational Position Indicator Controls

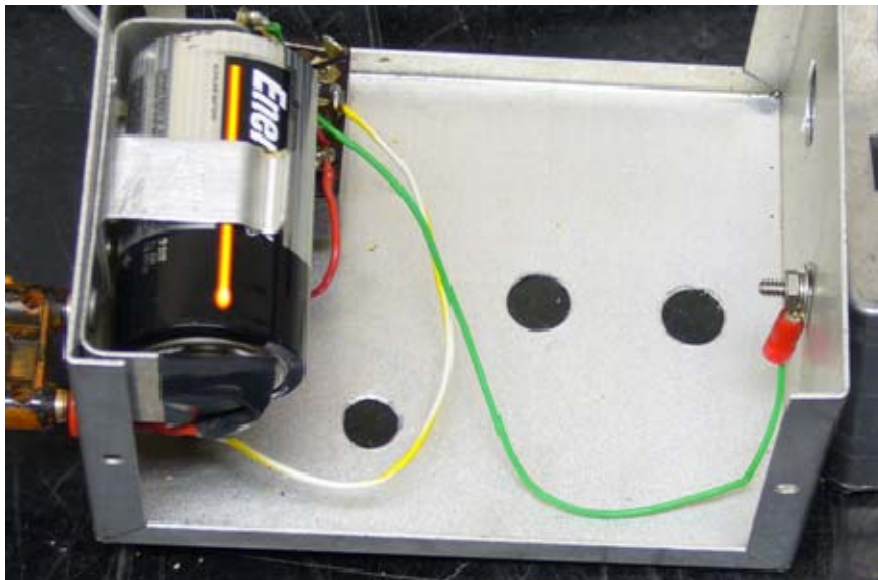


Figure 11. Rotational Position Indicator Battery

The mechanism transited a complete 360° circle in approximately 60 seconds. It was necessary to ensure two "over the top" voltage transitions from 0 to 1.5V were captured during the 93.75 second data stream. These transitions were used as the beginning and ending cut points in the data stream to ensure a complete continuous 360° of data were processed. To accomplish this, recording was started with the sensors in an orientation approximately $\pm 90^\circ$ to the 0° axis, corresponding to voltage readings of approximately 0.3V and 1.1V. It made no difference whether the mechanism rotated clockwise or counterclockwise. Both directions were used. Figure 12 shows the full 96-second voltage trace of a single data stream.

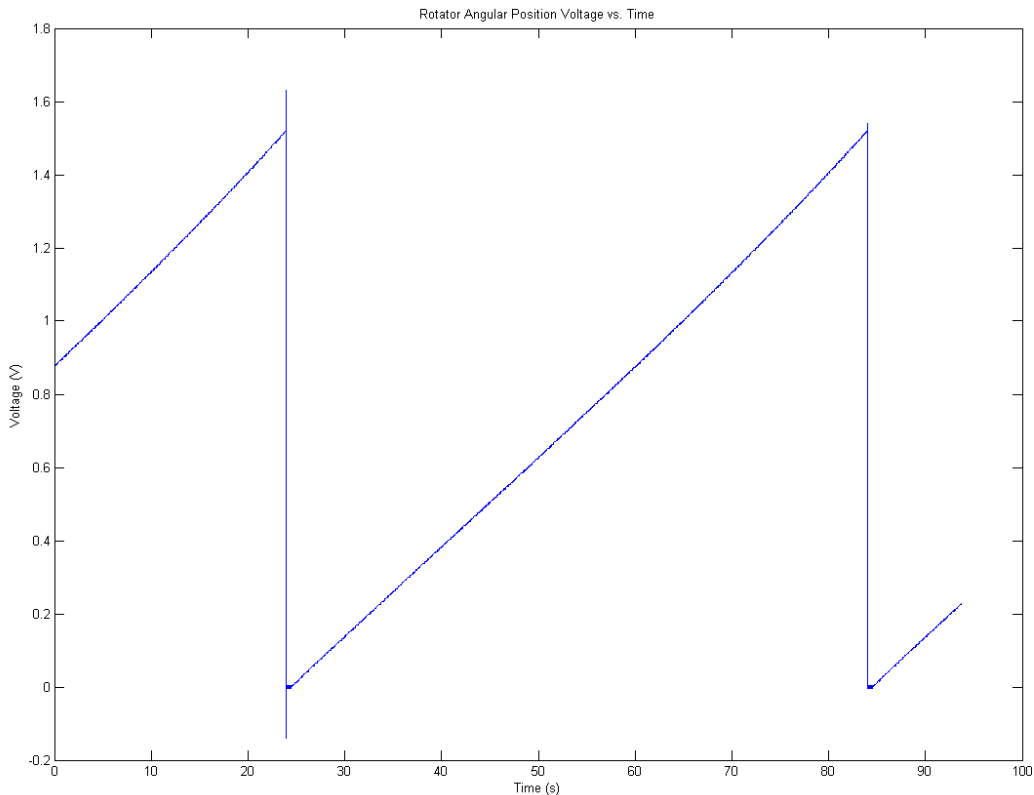


Figure 12. Complete Data Trace of the Voltage Position Signal

The voltage output of the rotator mechanism was used as a reference guide only. Response of the sensor itself was used to determine the actual polar positioning of the beam pattern during post-processing. The voltage output was not sufficiently calibrated to provide fine enough resolution to determine precise

positioning, nor was it absolutely repeatable after removal and reinstallation of the apparatus. The voltage signals from the potentiometer were oscillatory due to the bouncing of the wiper arm across the windings while the unit was in motion. Figure 13 displays the stepped transitions and oscillatory behavior. Figure 14 shows the voltage overshoots at the potentiometer wiper transition point. This prevented fine measurements or exact readings. There was also considerable backlash in the gear mechanism when changing rotation directions, clockwise or counter-clockwise. The voltage readings were only used to aid general alignment and to indicate the point to start data capture. Additionally, voltage overshoots above 1.5V occurred due to induction at the transition points as the wiper moved across the potentiometer gap.

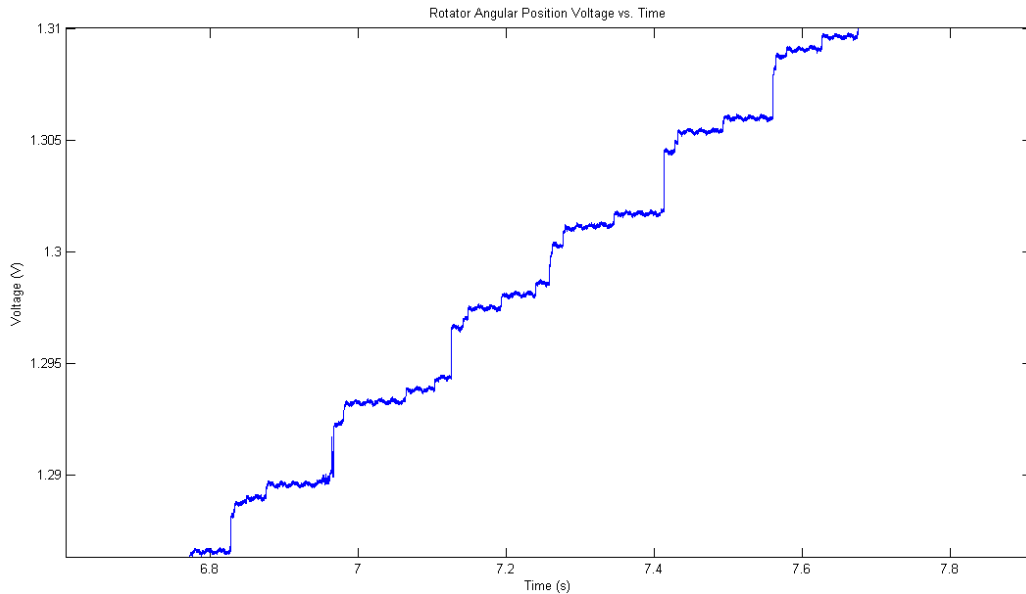


Figure 13. Close Up of the Voltage Position Signal with Stepped Transitions

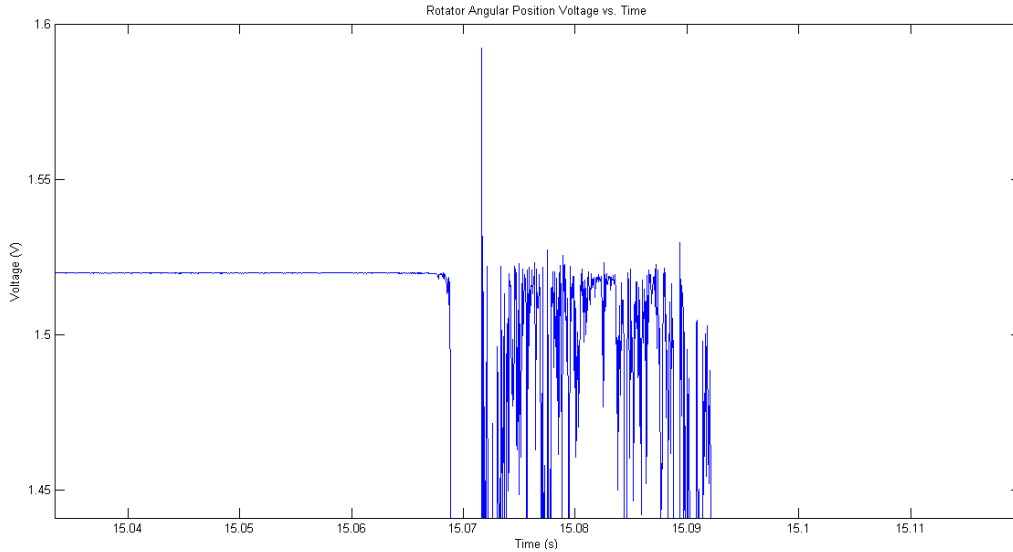


Figure 14. Noise and Overshoot of Voltage Position Signal

To negate this effect and the possibility of a partially discharged battery, a voltage level of 1.3V was used as the indicator for the transition cut point. The occurrence of any arbitrary voltage, sufficiently far away from the over the top transition points, could be used to delineate a complete circle of data. This was possible since, once in motion, the rate of rotation of the shaft was generally constant at the macro level. Thus, although 1.3V was used here, 0.3V or any other value in between could have worked equally well.

C. NOISE AND GROUNDING

During initial testing, unknown frequency components were discovered in the signals from the Microflown sensor. These frequency components were spaced approximately 60Hz apart, indicating they were noise from the electrical system. The power supply for the National Instruments 9172 chassis has a three prong grounded plug. The power supply for the Microflown has a two-prong plug. There was no common ground to the system. A grounding wire was fitted between the NI 9172 chassis external ground point and a screw on the exterior metal case of the Microflown signal conditioner. As an extra measure of protection, this wire was fitted to the ground plug of an ordinary three-prong plug

and grounded to Earth ground through the equipment power strip. A three prong grounded power supply could be substituted for the Microflown signal conditioner and would make a significant yet inexpensive upgrade. After the system was grounded the 60Hz noise was suppressed.

There are still low-level random frequency components in the signals used to process the polar plots as seen in the frequency domain plots, Figures 15 and 16.

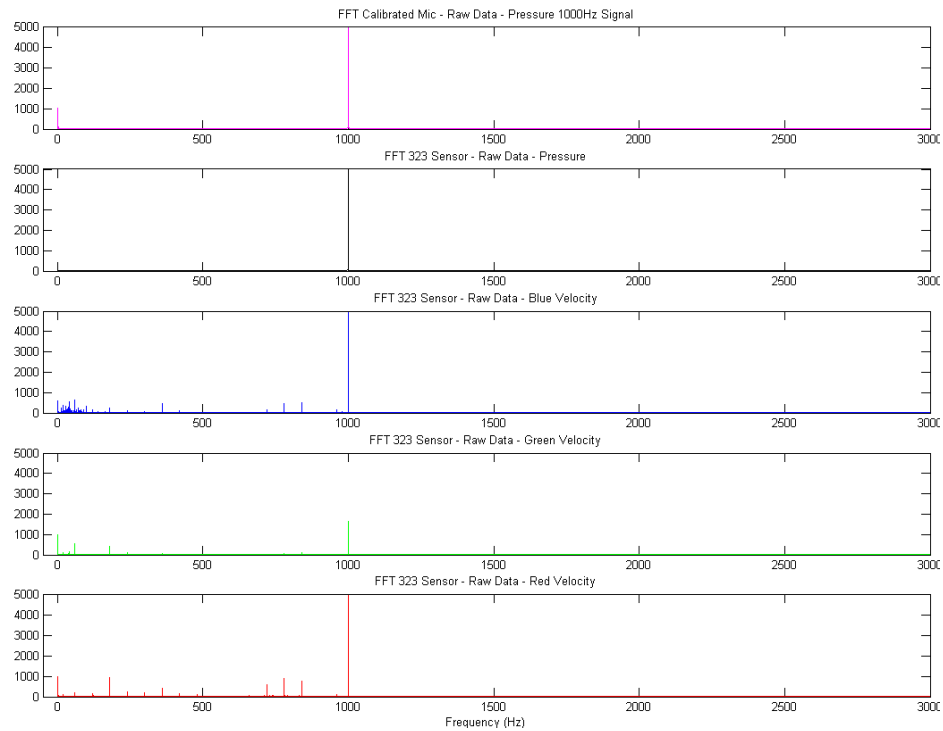


Figure 15. Sensor 323 Low Frequency Noise While Rotating

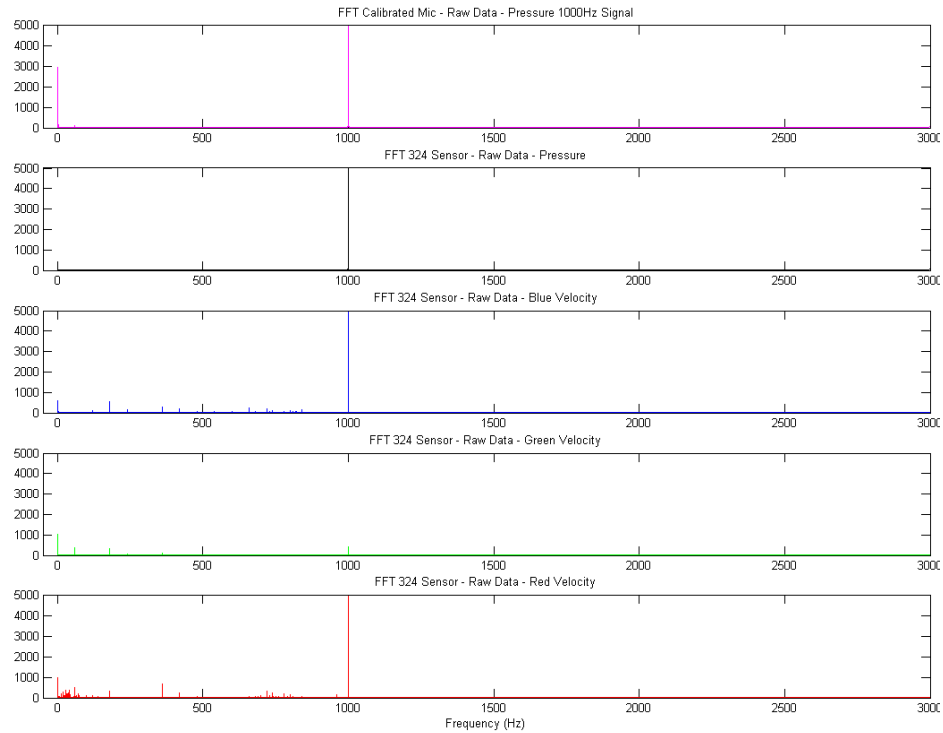


Figure 16. Sensor 324 Low Frequency Noise While Rotating

These components are not present in the pressure sensor traces but do exist in the velocity sensor traces. This noise is caused by motion and vibration of the rotator mechanism itself. When the apparatus moves, air flows over the velocity sensors, causing a temperature differential, which is recorded as voltage signals at random frequencies. It is not present when the mechanism is stationary. The amplitudes of these noise signals are so low compared to the source signal that they can be ignored. Windowing the signal around the signal of interest further increases the signal to noise ratio, reducing the effect of this noise.

V. BEAM PATTERNS OF THE MICROFLOWN SENSORS

A. METHOD

Characterization of the Microflown sensor began with determining the natural beam pattern of the individual sensors and their geometric relationship to one another. The Microflown was placed in a rotating fixture coaxial with the Aco microphone. The sensors were arranged such that the top surfaces of the two sensors faced each other with a gap of approximately 2cm. In order to minimize difference in path length effects from an off center mounting, the central axes of both sensors were aligned coaxial with the center of the rotating shaft.

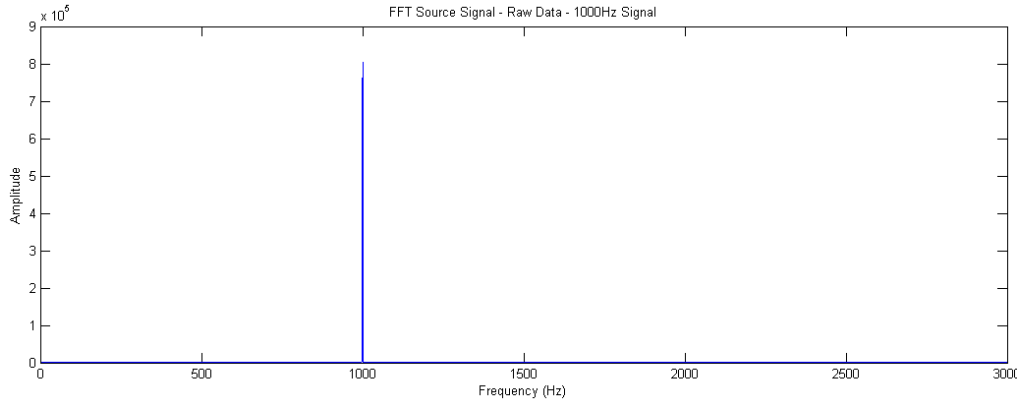


Figure 17. Source Signal Used for Polar Plot Testing

A source with frequency of 1kHz and amplitude of 1Vpp was fed into an amplifier, which in turn drove the Vifa loudspeaker. This signal is shown in Figure 17. Using an array aperture of $R = 5mm$, the far field was calculated as (Ziomek, 1994)

$$FF = \frac{\pi R^2}{\lambda} = \frac{\pi(0.005m)^2}{(0.340m)} = 0.0002m . \quad (\text{Eq. 5.1})$$

The loudspeaker was placed well in the far field at a distance of approximately 190cm. The rotator mechanism was energized and the data stream recorded.

B. POST-PROCESSING TO DETERMINE POLAR BEAM PATTERNS

Post-processing consisted of five steps.

1) A continuous 360° stream of data was extracted from the 93.75 second data stream. Cut points were determined using the recorded voltage trace as described in Chapter IV.B.

2) The data was de-meant and basebanded in the time domain. The baseband value was centered on the frequency of interest $f_0 = 1000\text{Hz}$ by multiplying the signal by $e^{i2\pi f_0 t}$.

3) The data was Fourier transformed into the frequency domain where the data was lowpass filtered to remove the sum frequency resulting from the basebanding.

4) The data was inverse Fourier transformed back to the time domain and plotted on a polar plot across a full 360° in θ . Plots were produced as both relative amplitude responses by sensor element and as normalized plots with all element response amplitudes equal in magnitude. The normalized plots aided sensor orthogonality examination.

5) Finally, the polar plots were aligned to the actual axes of the sensor using the minimum response of the red element. Since the +MRA of the red element is at $\theta = 90^\circ$, the minimum response of this element corresponds to the 0° axis. The minimum response point was determined by first finding the minimum response points in each half of the data stream, then taking the average of these two indices. The data was realigned using this index as the new 0° marker and re-plotted. Discontinuities in the polar plots appear at the seams, but these do not detract from the overall value of these plots. Using the discontinuities as guides, the range of angular rotation required to realign the plots can be readily observed. Figures 18 and 19 demonstrate how the start and stop points of the data capture (angular position of the apparatus when data capture is initiated) can differ widely between subsequent data collections.

Regardless of these differences, the voltage position indicator and realignment routine used to determine the cut points and rotation produce similar final results. Comparing the before and after plots of both sensors, the plot for sensor 323 on the left required a much larger degree of realignment than the plot for sensor 324 on the right, 130° vs. 5° . Thus, the data capture can begin at an arbitrary angle and still deliver satisfactory results with this method.

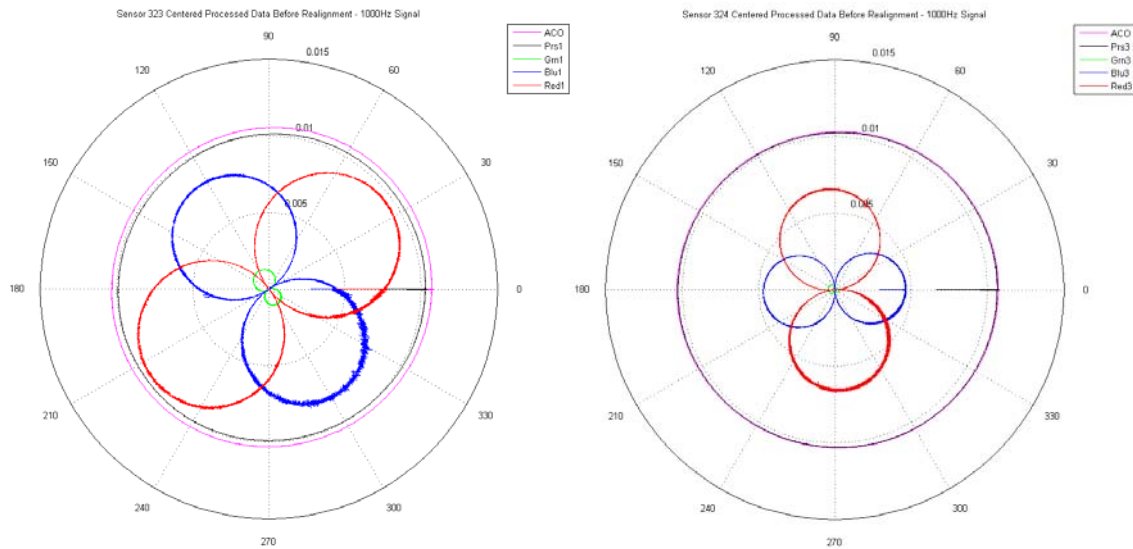


Figure 18. Uncorrected Polar Plots

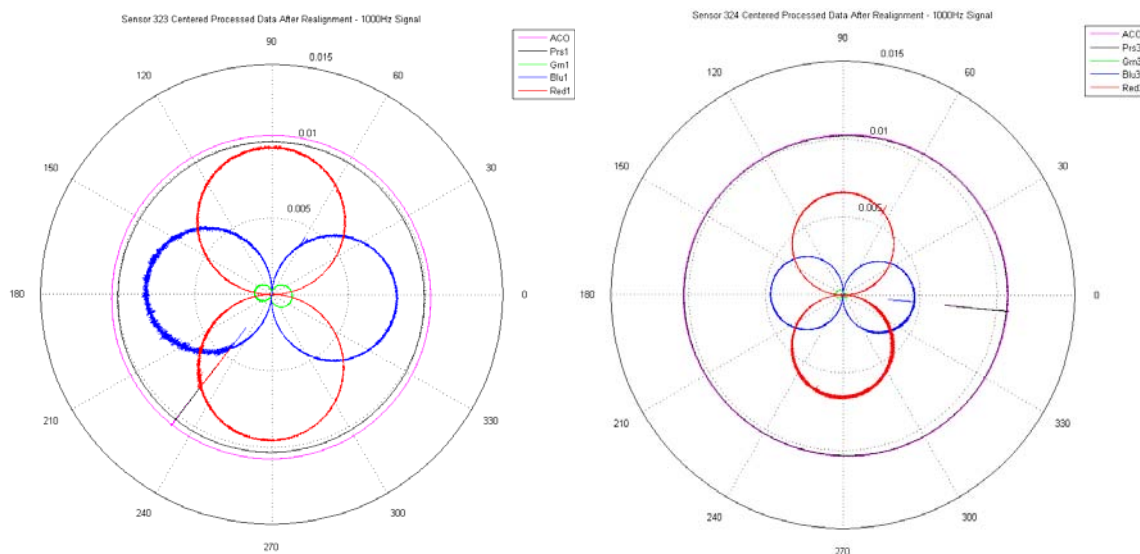


Figure 19. Realigned Polar Plots

C. RESPONSE AND BEAM PATTERNS OF THE INDIVIDUAL VELOCITY ELEMENTS

The polar plots show the relative amplitude responses of the various elements over the complete 360° of rotation. The expected “figure eight” beam pattern of a directional element can readily be identified with the three velocity elements. Similarly, the circular beam pattern of the omni-directional pressure elements can also be readily identified. Since the MRA of the green element is perpendicular to the plane of rotation it has nearly no response to the test signal at all angles during the test. In the normalized plots, Figure 20, its response is greatly exaggerated. Also, note the nominal orthogonality of the three velocity sensors with their individual beam patterns at right angles to one another. Due to difficulty in apparatus construction, no rotation was performed in the ϕ direction for these tests. Therefore, the MRA response of the green element was not actually tested. However, with the information from these polar plots and the static test data collected for the transfer functions in Chapter VI, a reasonable conclusion can be drawn regarding its orthogonality and sensitivity relative to the other elements.

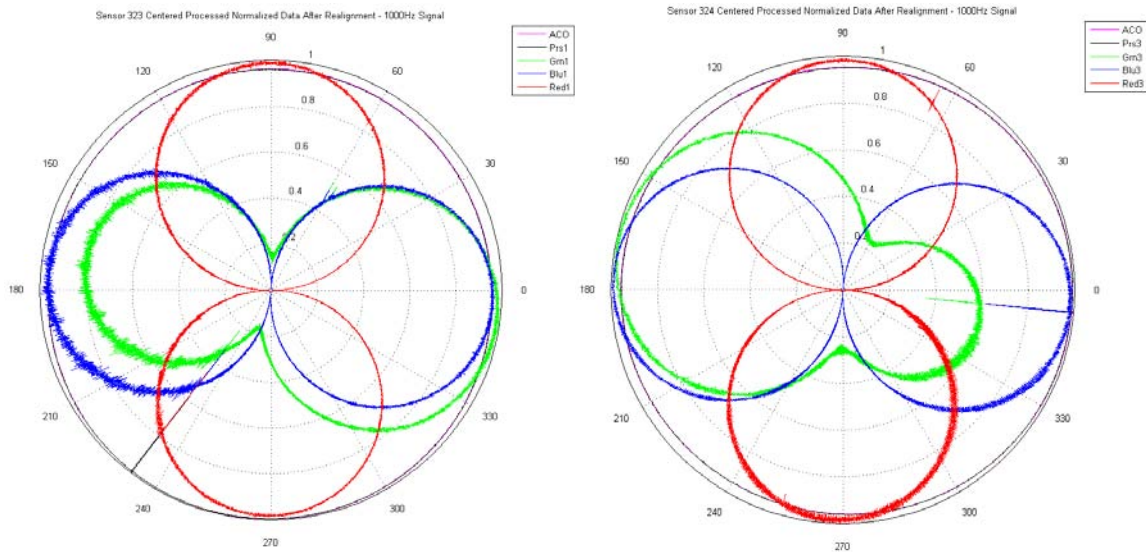


Figure 20. Realigned Normalized Polar Plots

D. ORTHOGONALITY OF THE VELOCITY ELEMENTS

The individual velocity elements are nominally orthogonal to one another, but they are not absolutely orthogonal. This is most readily apparent in the realigned normalized polar plots in Figure 20.

Since the plots were aligned using the red element as the reference, the red element plots display as perfectly orthogonal on the polar plots. The blue element plots are not perfectly aligned however. In both figures, the blue element beam patterns are rotated approximately 5° from perpendicular relative to the red plots. The maxima of the blue element responses lie at approximately 5° and 185° rather than 0° and 180° as expected. Similarly, the green element and blue element plots do not exactly align. This latter point is most readily apparent on the plot for sensor 324, where the green element plot is displaced an additional 10° from the blue element beam pattern, or a total of 15° from the absolute reference.

The non-orthogonality of the velocity elements is partially mitigated by the slope of the response near the maxima of the beam patterns. The amplitude of the response in the region $\pm 10^\circ$ on either side of the positive and negative MRA axes is nearly constant, even more so in the region $\pm 5^\circ$ of the MRA axes. Because of this, no provision was made to account for the lack of absolute orthogonality of the velocity elements in the beamformer.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CHARACTERIZING THE RESPONSE OF THE VELOCITY ELEMENTS AND CALCULATING TRANSFER FUNCTIONS

A. METHOD

The next stage of characterization of the Microflown sensors was to calculate relative transfer functions between each element in the array to a common reference. All of the data for the transfer functions was taken with the sensors in their final positions within the array. This was done to preserve the relative positions in three dimensions to preclude recalculation of the transfer functions upon successive data collection efforts.

The hybrid linear array was constructed with the Aco microphone at the center, the Microflown 323 sensor 172mm away in the positive z direction and the Microflown 324 sensor 172mm away in the negative z direction. The Aco was designated as the origin of the coordinate system for all calculations of distances and angles. The response axes of the Microflown sensors were aligned relative to the axes of the array. To accomplish this, the array was mounted at $\theta = 90^\circ$. Using an array aperture of $R = 340mm$, corresponding to the total length of the array, the far field was calculated as (Ziomek, 1994)

$$FF = \frac{\pi R^2}{\lambda} = \frac{\pi (0.340m)^2}{(0.340m)} = 1.07m. \quad (\text{Eq. 6.1})$$

A 1kHz source was placed in the far field. A laser level was used to ensure the all sensors were in the same plane and also co-planar with the midpoint of the source driver. An oscilloscope and digital voltmeter were used to monitor the response amplitude of the blue elements. The Microflown sensors were rotated about their longitudinal axes within their mounting holes in the fixture until the minimum response was determined. The minimum response of the blue elements corresponds to $\theta = 90^\circ$. This aligned the MRA of the blue elements to $\theta = 0^\circ$, the longitudinal axis of the fixture and the z axis of the coordinate system. The fixture was held fixed while adjusting the sensors in order to preserve its absolute aspect relative to the source. Figure 21 shows

aligning the array plane with the centerline of the loudspeaker using a laser level. Figure 22 shows the equipment setup and procedure to align the sensors in the array.



Figure 21. Ensuring the Array is Coplanar with Source, Note Foam Cover

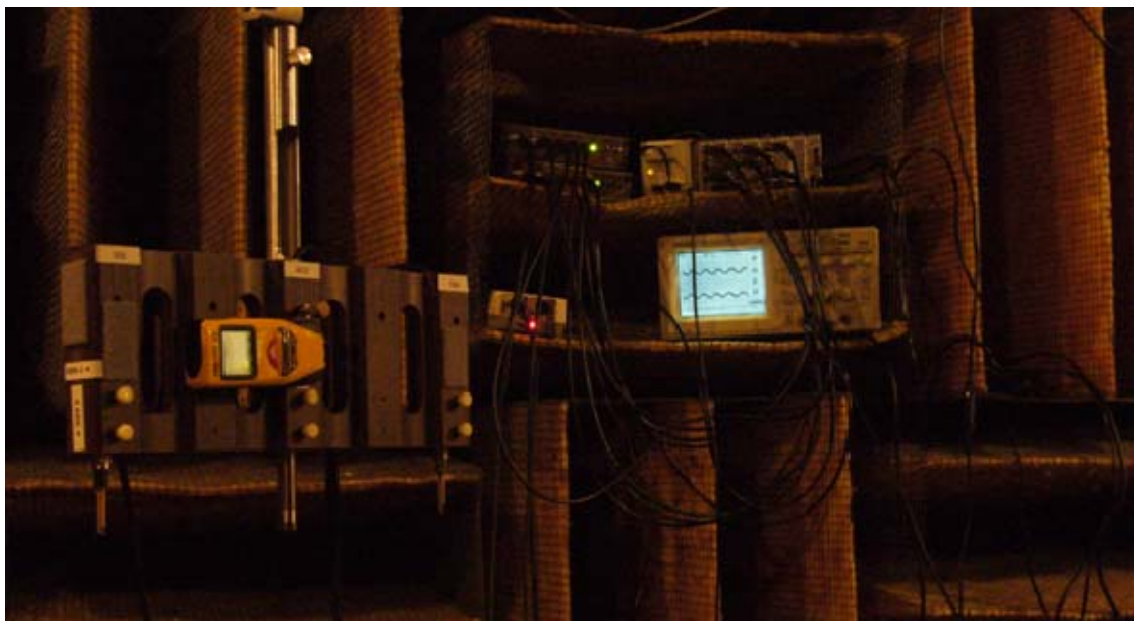


Figure 22. Aligning The Sensors in the Array Using an Oscilloscope

Once the sensors were aligned, their positions were secured with nylon setscrews. Witness marks were made along paper tape markers on the fixture and sensors in order to provide a means of monitoring the sensors' positions in

the fixture, and as a means to restore the sensor position without recalibration should they need to be removed from the fixture.

The transfer function signal was a white noise source with an amplitude of 5Vpp. This signal was then fed into the amplifier that drove the Vifa loudspeaker with an average amplitude of ~7Vpp. The source and receiver spectra are shown in Figures 23 and 24. The loudspeaker was placed in the far field at distances over 2m for each data collection series. The measurements for the transfer functions were taken in static positions corresponding to the MRAs of each of the three velocity elements.

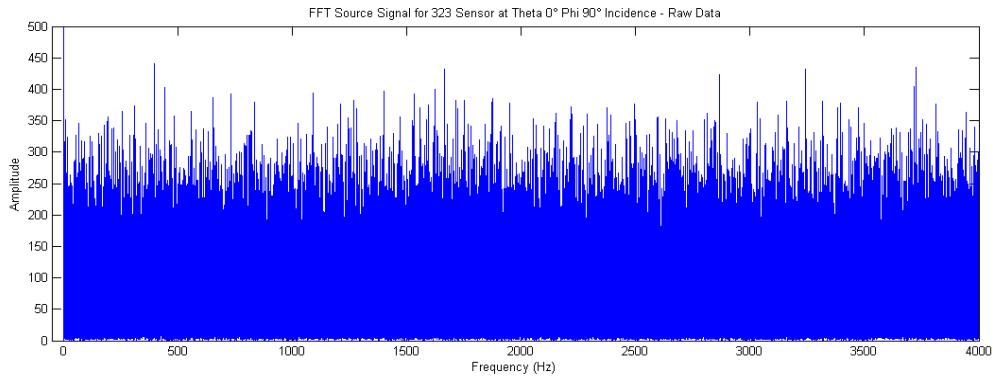


Figure 23. Source Signal Used for Determining Transfer Functions

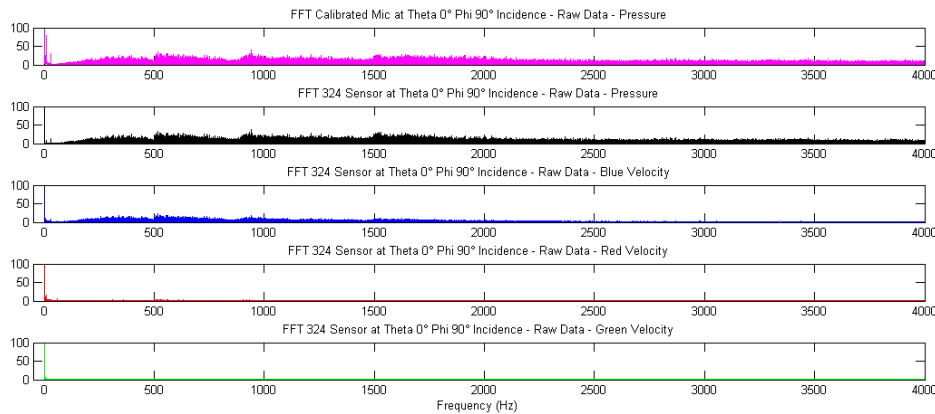


Figure 24. Frequency Components By Element With Array Aligned to Blue MRA

Velocity Elements Angles and MRAs			
Velocity Element	θ	ϕ	MRA Axis
Green	90°	90°	x
Red	90°	0°	y
Blue	0°	90°	z

Table 2. Angular Alignments by Element Used in Determining Transfer Functions

Table 2 above shows the array orientations for transfer function data capture by relevant velocity element. Three sets of data were collected in each orientation. Each dataset was processed independently and the results for each element were averaged to produce the final transfer functions. A transfer function was obtained for each Microflown pressure element relative to the Aco microphone. Transfer functions for the individual Microflown velocity elements were obtained relative to their respective onboard pressure elements. The raw transfer function plots, Figures 25-32 display all three data sets plotted together. Note the close agreement and repeatability of the various data sets within each plot.

B. CALCULATING THE TRANSFER FUNCTIONS

Post-processing consisted of four steps. In each of these steps, the data was analyzed in chunks of 65,536 data points. Hanning windows were applied to the time domain data, and successive chunks of data used an overlap equal to one-half the length of the data chunk. The resulting complex transfer function length was 32,768 (2^{15}) corresponding to a maximum usable Nyquist frequency of 12.8kHz. The steps were:

- 1) A Magnitude Squared Coherence Estimate was calculated between the two respective elements using *mscohere*. This function was used to verify sufficient signal to noise level was present in order to validate calculation of the transfer functions.

2) A Cross Power Spectral Density estimate using Welch's method was calculated between the two respective elements using *cpsd* .

3) A Power Spectral Density estimate using Welch's method was calculated for the reference element using *pwelch* .

4) The final transfer function was calculated by dividing the Power Spectral Density estimate of the reference element by the Cross Power Spectral Density estimate calculated between the two respective elements.

Note the phase of the red and blue elements compared between the two sensors in the plots, Figures 27 and 28 and Figures 31 and 32 respectively. This indicates they have opposite +MRA axis polarities even though they are installed in a physically identical orientation in the fixture with respect to the relative position of each element. This is likely caused by the different polarities of the element lead wires inside the casing during manufacture.

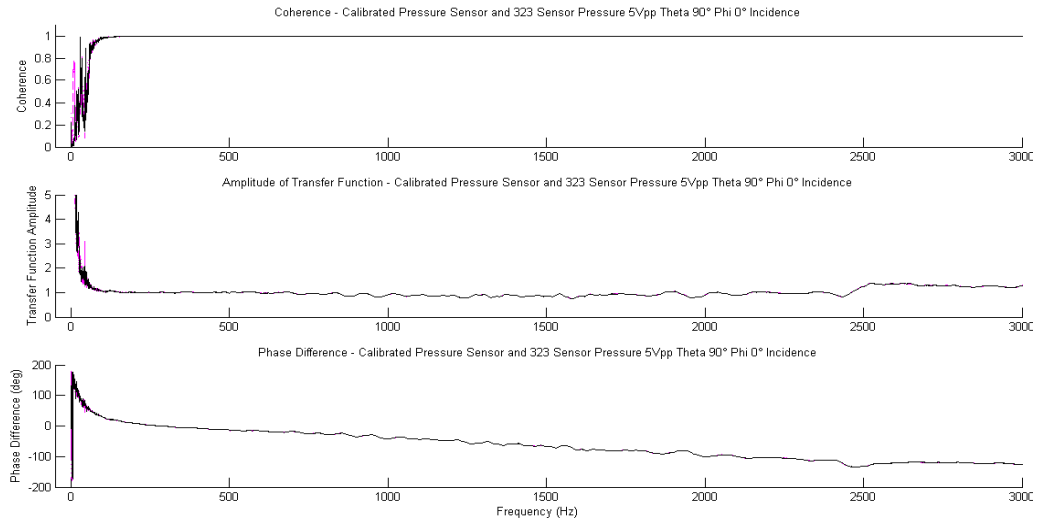


Figure 25. Raw Pressure Element Transfer Functions for Sensor 323

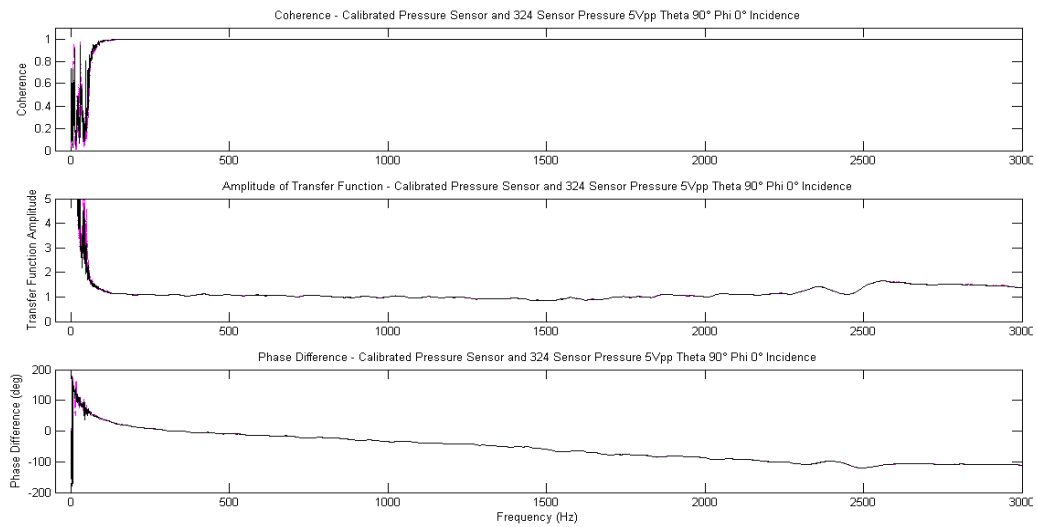


Figure 26. Raw Pressure Element Transfer Functions for Sensor 324

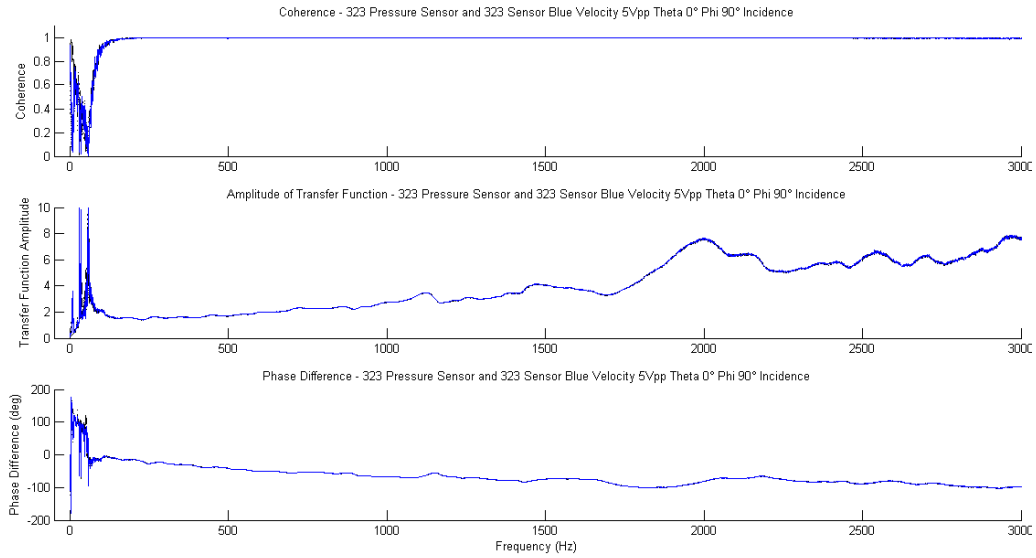


Figure 27. Raw Blue Velocity Element Transfer Functions for Sensor 323

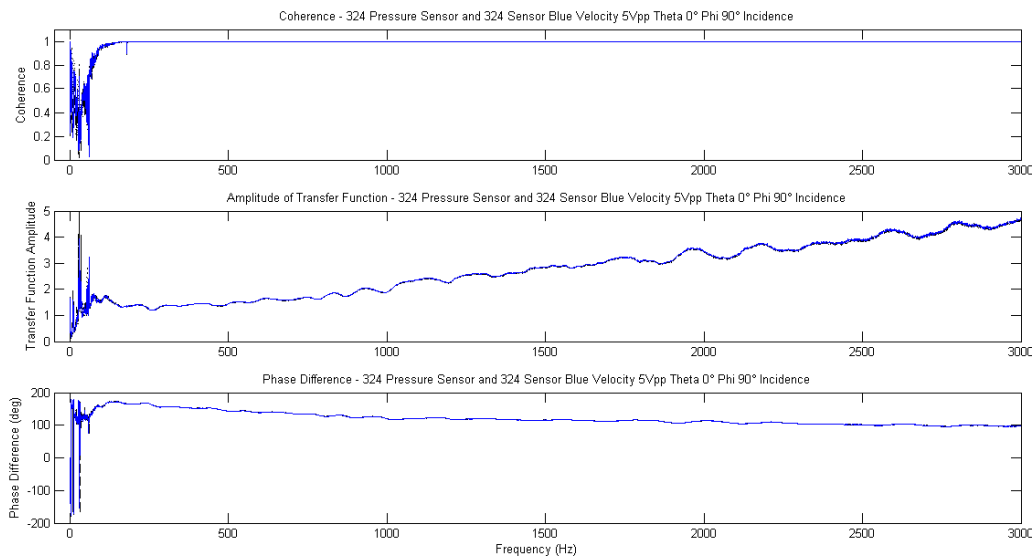


Figure 28. Raw Blue Velocity Element Transfer Functions for Sensor 324

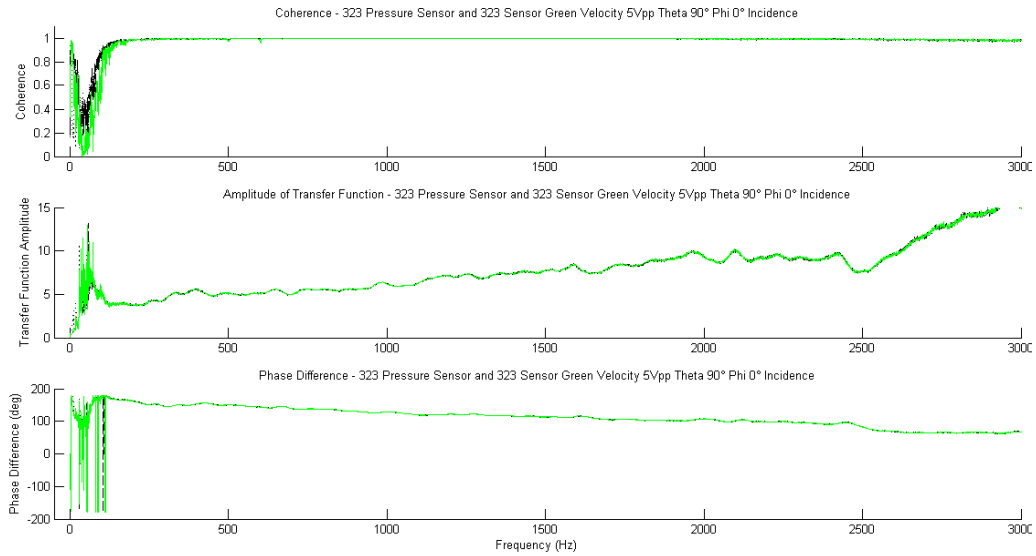


Figure 29. Raw Green Velocity Element Transfer Functions for Sensor 323

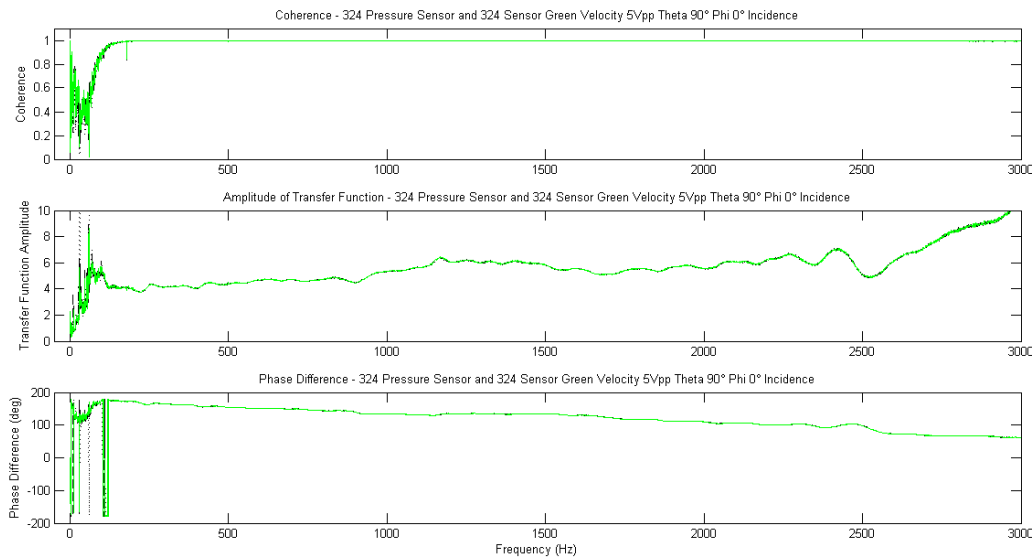


Figure 30. Raw Green Velocity Element Transfer Functions for Sensor 324

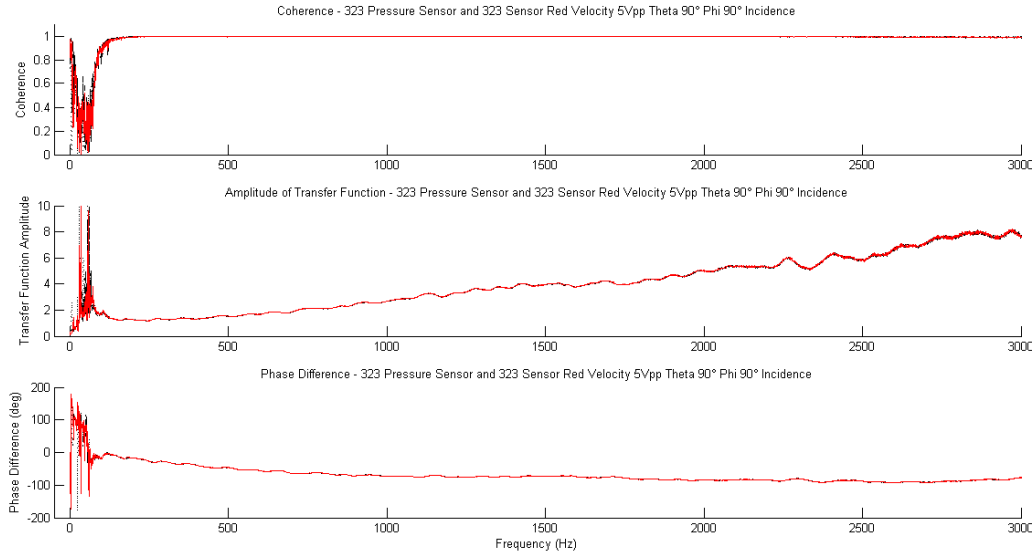


Figure 31. Raw Red Velocity Element Transfer Functions for Sensor 323

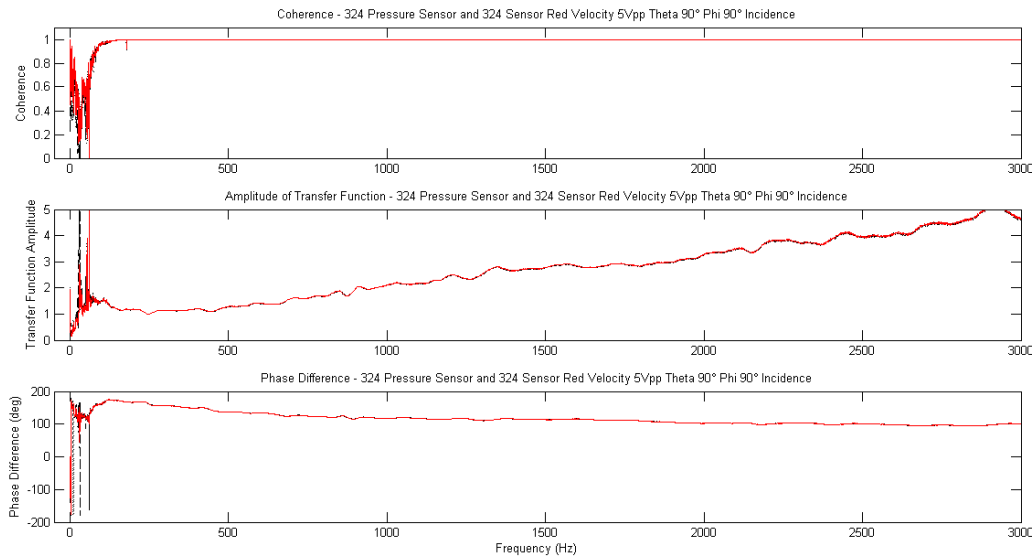


Figure 32. Raw Red Velocity Element Transfer Functions for Sensor 324

Once all of the individual transfer functions were calculated, the three transfer functions for each element were averaged into the final element transfer function. The beamformer uses the Aco microphone as a common reference to determine the composite beam pattern. In order to reference the velocity element signals to the Aco microphone in the beamformer, the transfer functions of the individual velocity elements relative to their respective pressure elements were multiplied by the transfer function of the onboard pressure element relative

to the Aco microphone. Figures 23-40 show the final averaged and pressure multiplied transfer functions in the frequency region of interest. Since the transfer functions are calculated in the frequency domain, element-by-element multiplication can be used in place of convolution as required in the time domain. Table 3 displays the final transfer functions in terms of the transfer functions correlating the velocity elements to their respective onboard pressure elements, e.g. \hat{H}_{12} , and the transfer functions correlating the onboard pressure element with the central Aco microphone, e.g. \hat{F}_{12} . Care must be taken when changing the transfer functions and data signals from columnar to row format. The non-conjugate transpose function must be used to avoid changing the phase of the transfer functions. This same caution applies in the beamformer any time matrices are reformatted or transposed. The negative effects of inverted phase calculations cannot be understated. The apparent discontinuities in the transfer function phase plots are merely a function of the calculation "going over the top" when reaching an angular limit of $\pm\pi$. This has no effect of the calculation in the beamformer.

Sensor	Element			
	Pressure	Blue	Green	Red
323	\hat{F}_{12}	$\hat{H}_{t12} = \hat{H}_{12} \hat{F}_{12}$	$\hat{H}_{t13} = \hat{H}_{13} \hat{F}_{12}$	$\hat{H}_{t14} = \hat{H}_{14} \hat{F}_{12}$
Aco	1	0	0	0
324	\hat{F}_{32}	$\hat{H}_{t32} = \hat{H}_{32} \hat{F}_{32}$	$\hat{H}_{t33} = \hat{H}_{33} \hat{F}_{32}$	$\hat{H}_{t34} = \hat{H}_{34} \hat{F}_{32}$

Table 3. Final Transfer Function Calculations by Element

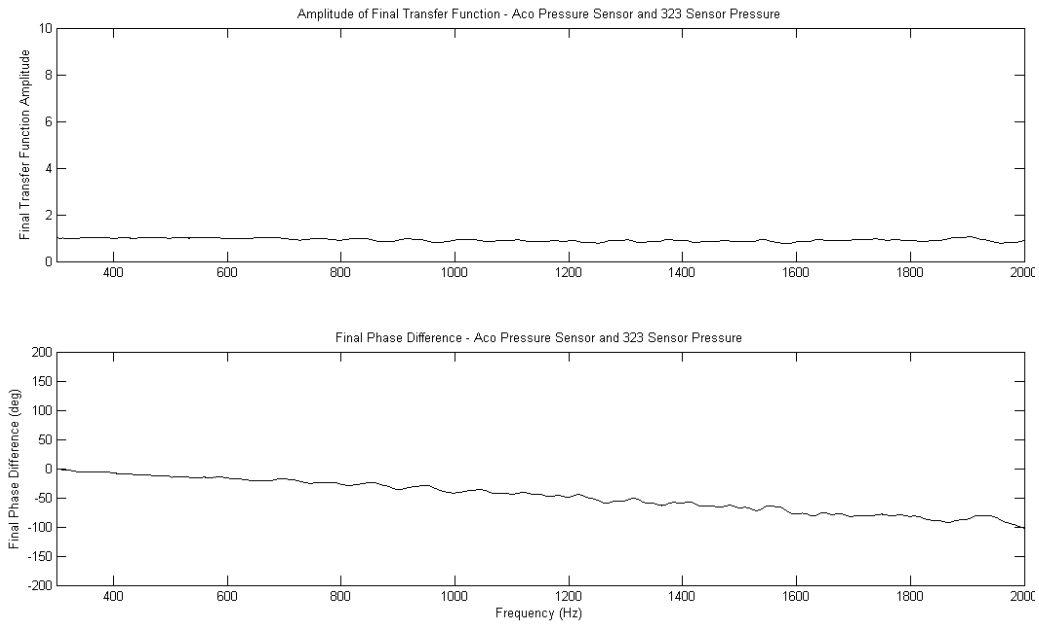


Figure 33. Final Averaged Pressure Element Transfer Function for Sensor 323

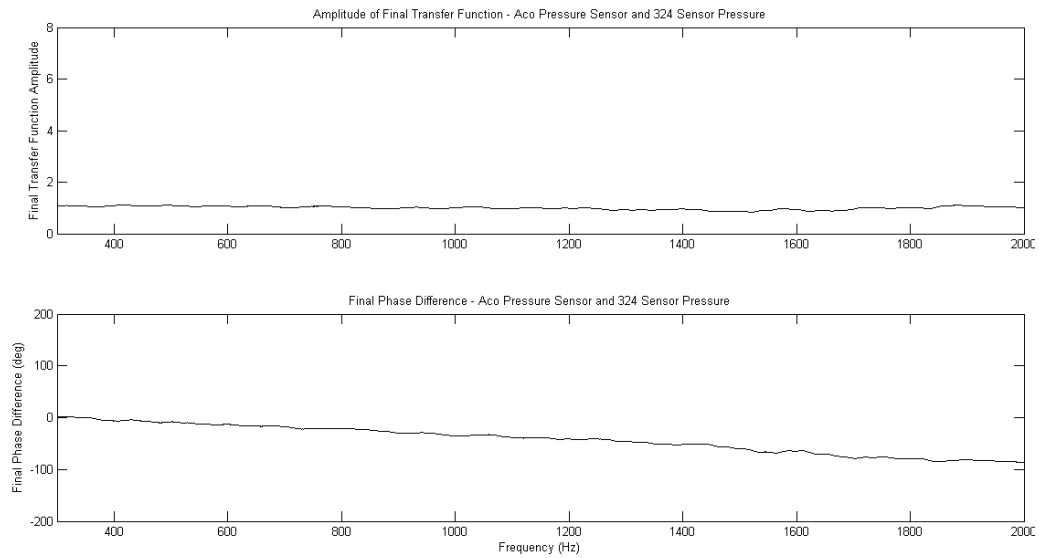


Figure 34. Final Averaged Pressure Element Transfer Function for Sensor 324

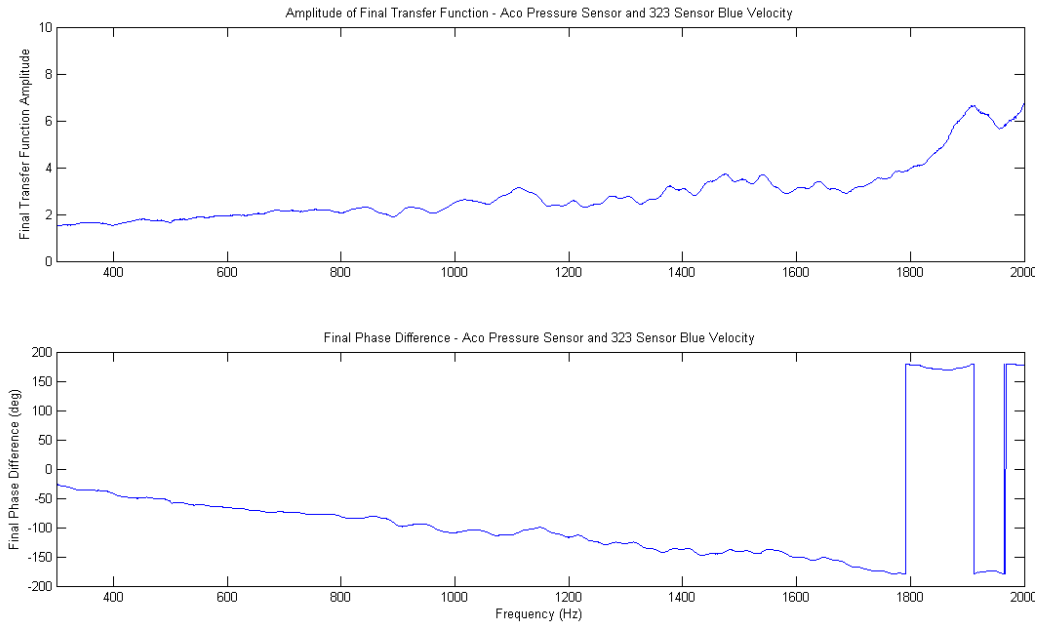


Figure 35. Final Averaged Blue Velocity Element Transfer Function for Sensor 323

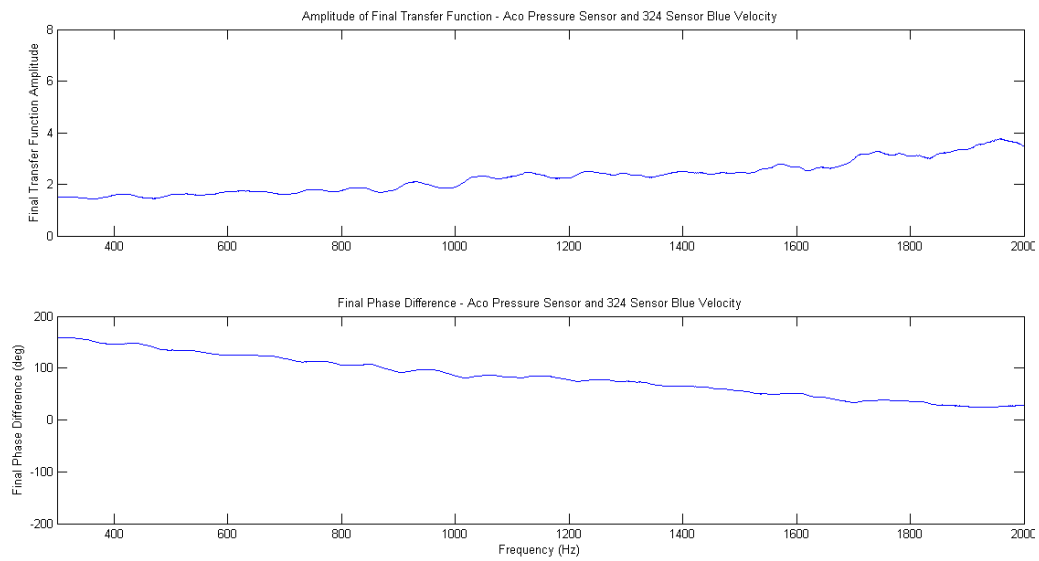


Figure 36. Final Averaged Blue Velocity Element Transfer Function for Sensor 324

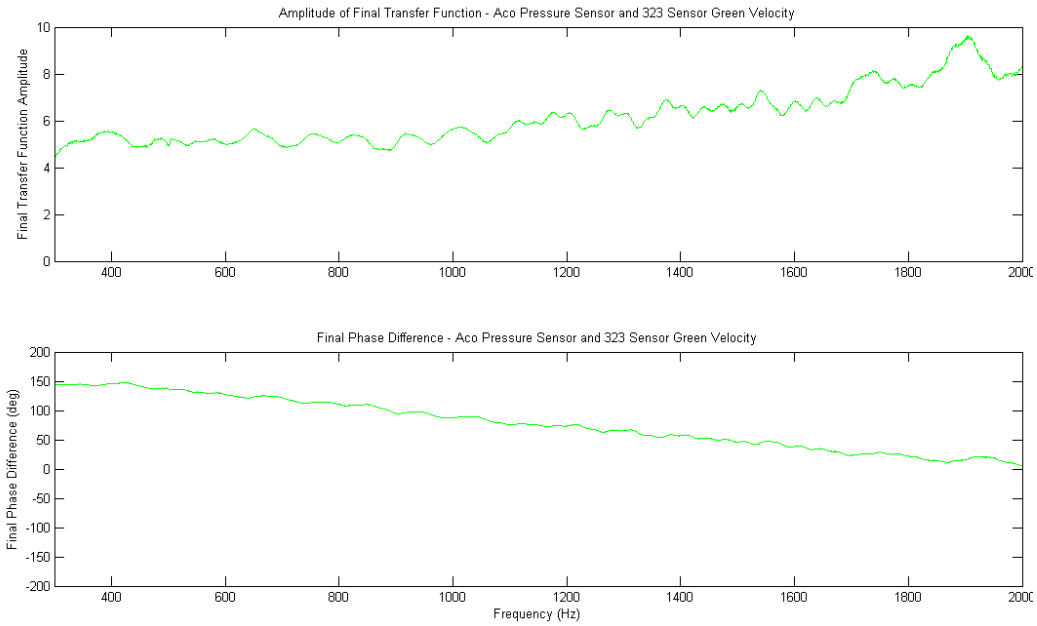


Figure 37. Final Averaged Green Velocity Element Transfer Function for Sensor 323

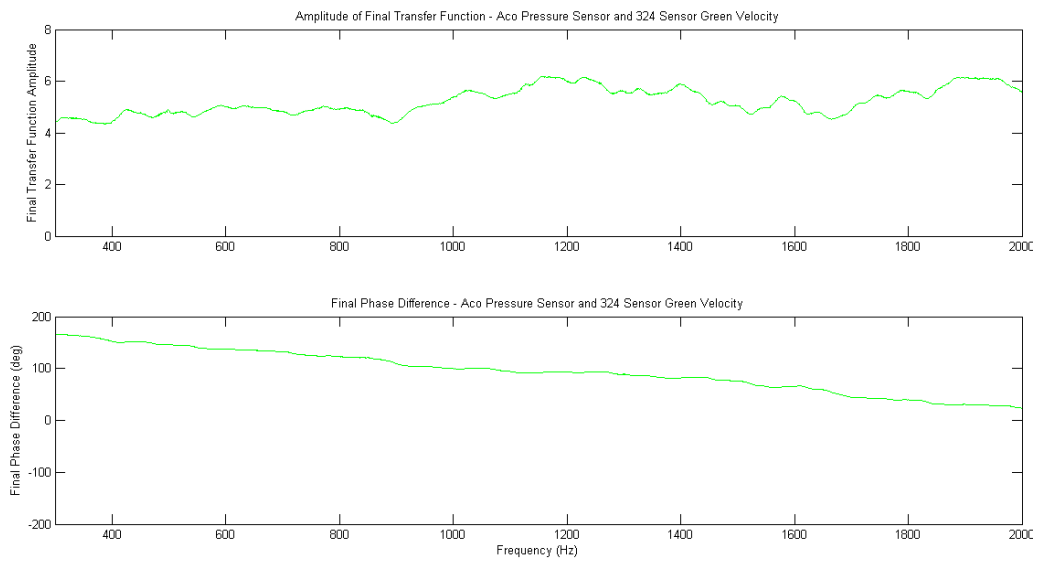


Figure 38. Final Averaged Green Velocity Element Transfer Function for Sensor 324

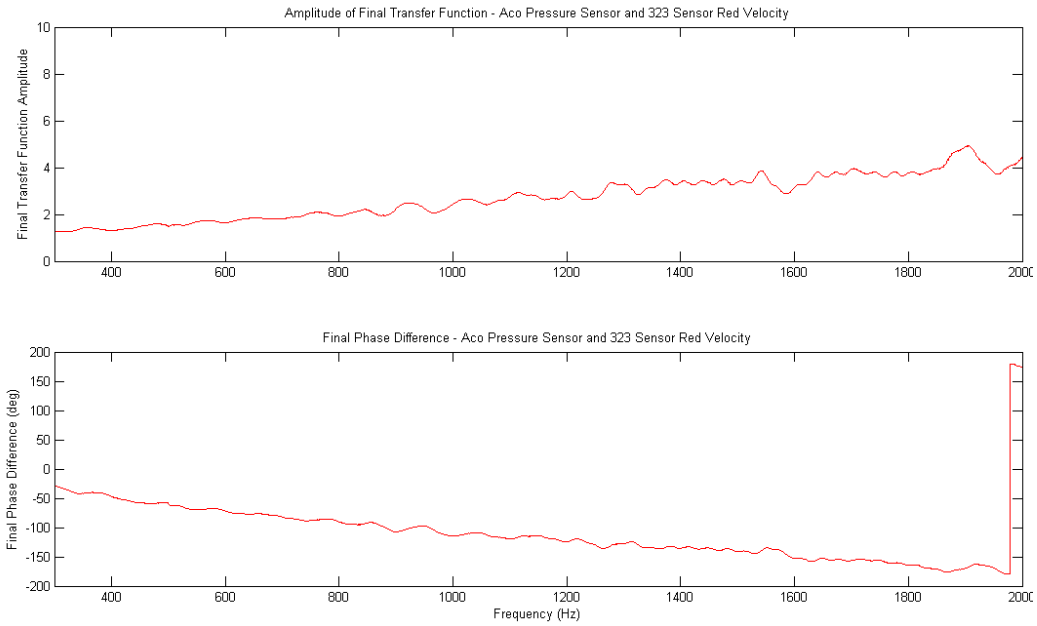


Figure 39. Final Averaged Red Velocity Element Transfer Function for Sensor 323

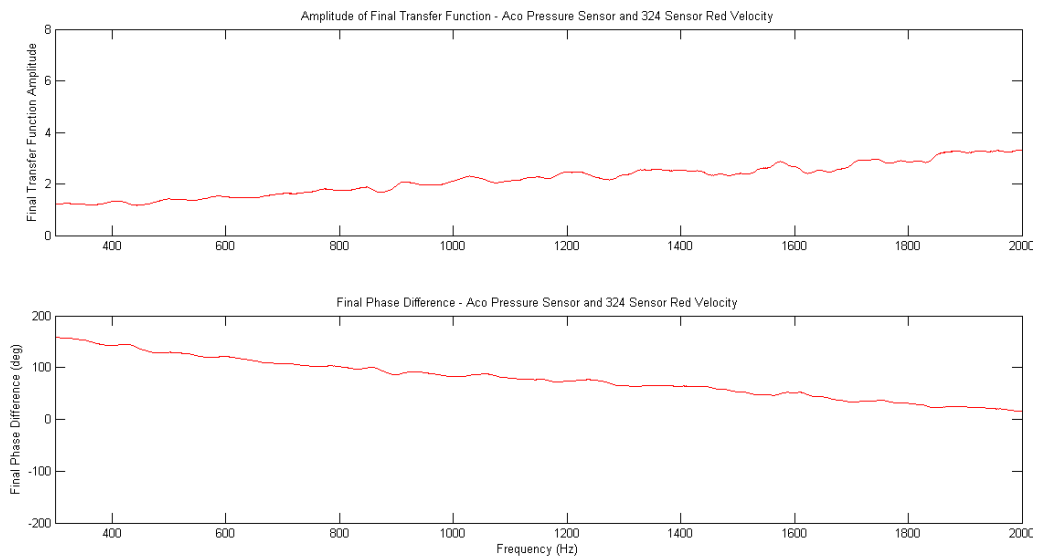


Figure 40. Final Averaged Red Velocity Element Transfer Function for Sensor 324

C. COHERENCE

All datasets for all elements in all orientations showed excellent coherence above 300Hz. This limit was due largely to the inability of the midsize loudspeaker to reproduce low frequencies. This frequency was used as the absolute lower limit for testing. The Microflow sensors showed good response in the low frequency range down to approximately 100Hz. If new transfer functions were calculated using a low frequency driver, the coherence and beamformer performance would likely improve in this region.

D. AMPLITUDE

For each element and for each data set, the amplitude of the transfer functions agreed extremely well above the 300Hz point. A quick test of the validity of the transfer functions was conducted by using a signal of 1kHz and plotting the amplitude of the Aco reference element versus the corrected amplitudes of the pressure and velocity element in each orientation. All three velocity element amplitudes showed an increase, with the red element showing the least augmentation in this test, and both pressure elements agreed closely. These results were repeatable with all velocity elements along their respective MRAs. Figures 41-43 show representative amplitude corrections.

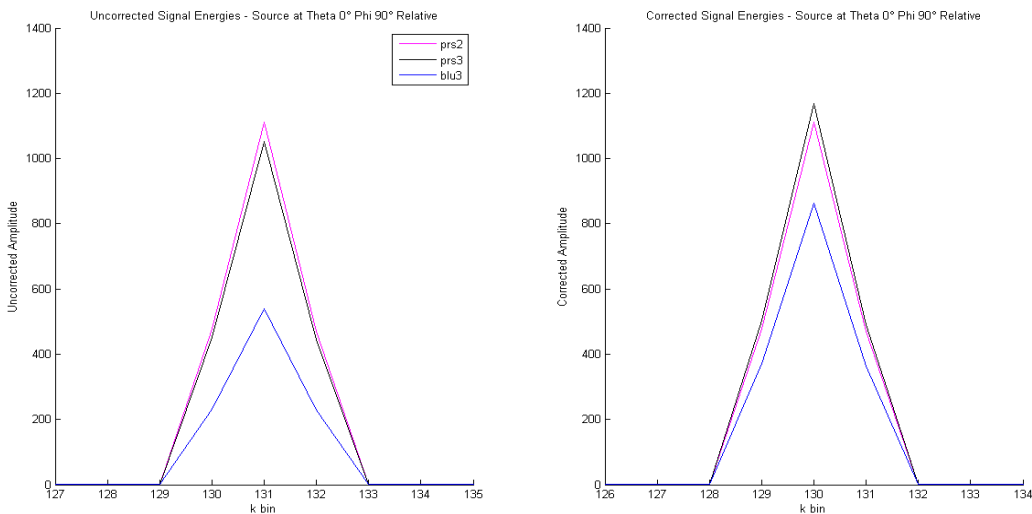


Figure 41. Raw and Corrected Signal Energies of a 1kHz Signal, Sensor 324 Blue

Element

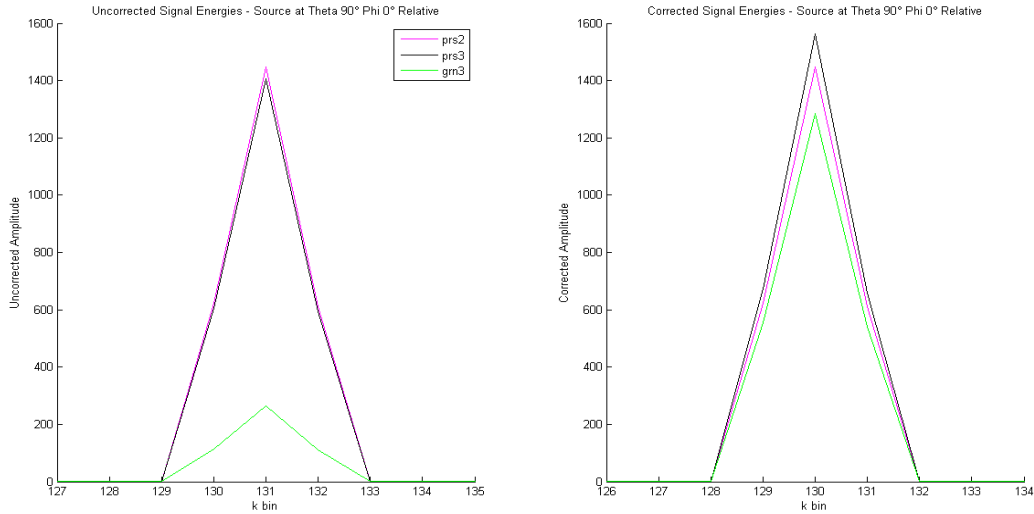


Figure 42. Raw and Corrected Signal Energies of a 1kHz Signal, Sensor 324 Green Element

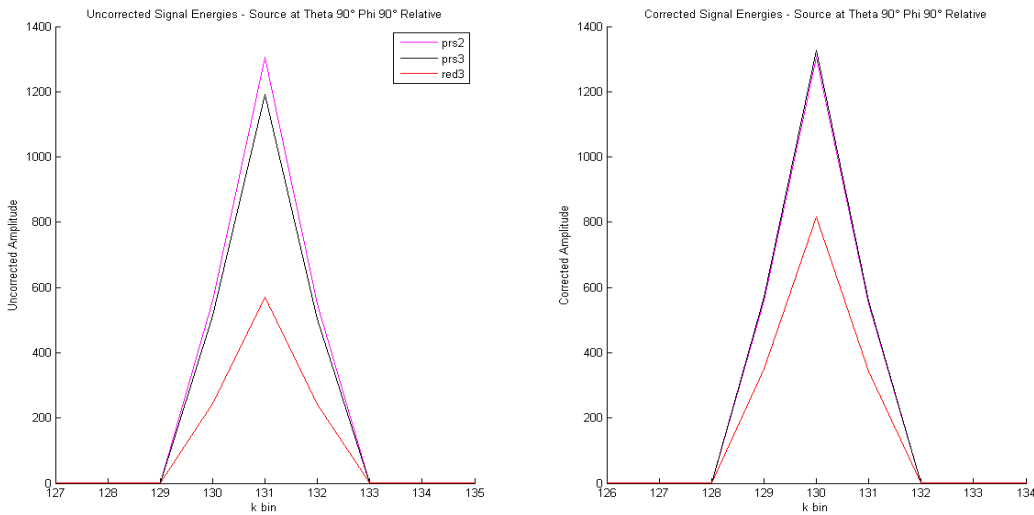


Figure 43. Raw and Corrected Signal Energies of a 1kHz Signal, Sensor 324 Red Element

An interesting result was observed in all velocity element transfer functions. The amplitude was not flat with frequency. There was considerable repeatable structure in them. Some of the structure occurred over a region of only a few tens of Hertz. The general shape of the blue and red elements was similar, but the green element was quite a bit different. This agrees roughly with the Microflown calibration documents. Another factor affecting the green element

may have been the existence of the apparatus itself in the -MRA plane. Even though the fixture was covered with foam rubber, the sensor elements only extended approximately 4cm from the top edge. It is possible some reflections were occurring and altering the green element trace.

E. PHASE

Similar to the amplitude traces, the phase of the transfer functions agreed extremely well above the 300Hz point for each element and for each data set. These results were also repeatable. The corrected phase responses of the elements exhibited better linearity and the shape and values of all three were similar. The -MRA plots for the blue and red elements showed the expected 180° phase shift. No -MRA data was recorded for the green element due to physical limitations and presence of the fixture mount itself in the green -MRA direction.

F. RESPONSE TO VARIOUS SIGNAL AMPLITUDES

Tests were conducted varying the source signal amplitude keeping the sensor in a fixed position to determine whether the velocity element responses were amplitude dependent. The data was recorded using signal amplitudes of 1Vpp, 2Vpp, 4Vpp and 5Vpp on the MRAs of both the blue and red elements separately.

The 1Vpp signal did not exhibit adequate coherence in order to validate the transfer function. The 2Vpp showed nearly the same coherence as the higher level signals. The coherence of the 4Vpp and 5Vpp signals was adequate and agreed well. From this data, it is reasonable to conclude a minimum test signal amplitude of 3Vpp is required. All transfer functions were computed with 5Vpp signals. Figures 44-49 show the results of these tests.

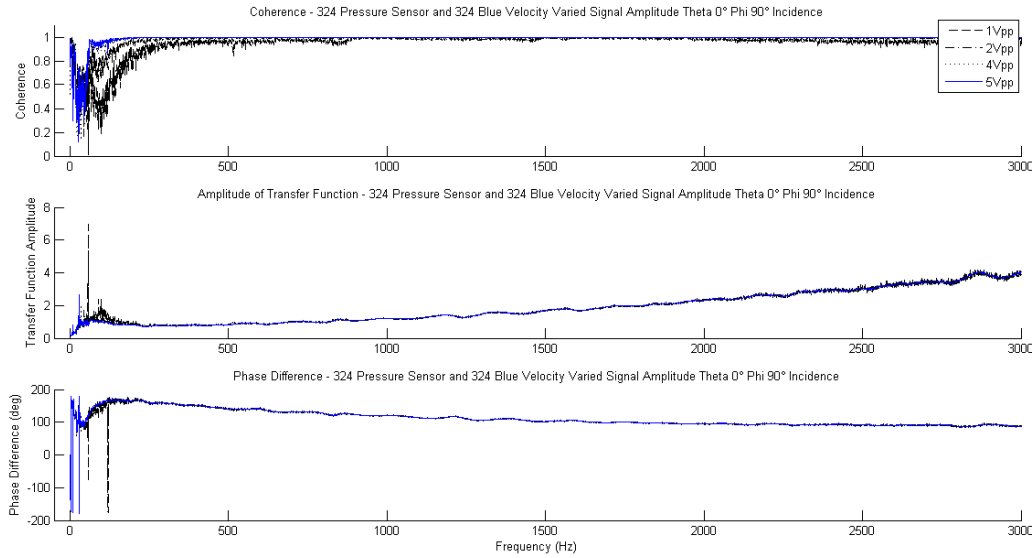


Figure 44. Sensor 324 Blue Element Transfer Function at Various Signal Amplitudes

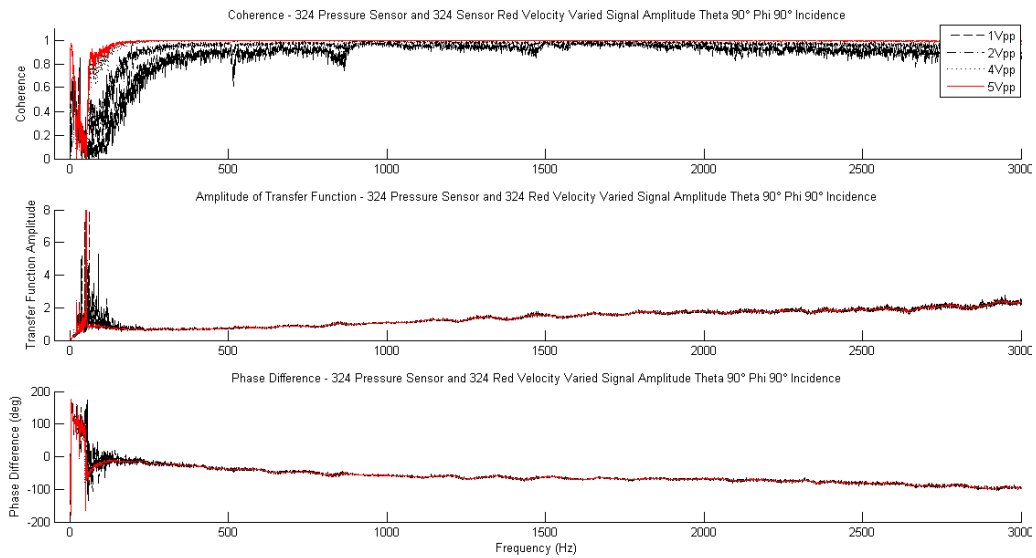


Figure 45. Sensor 324 Red Element Transfer Function at Various Signal Amplitudes

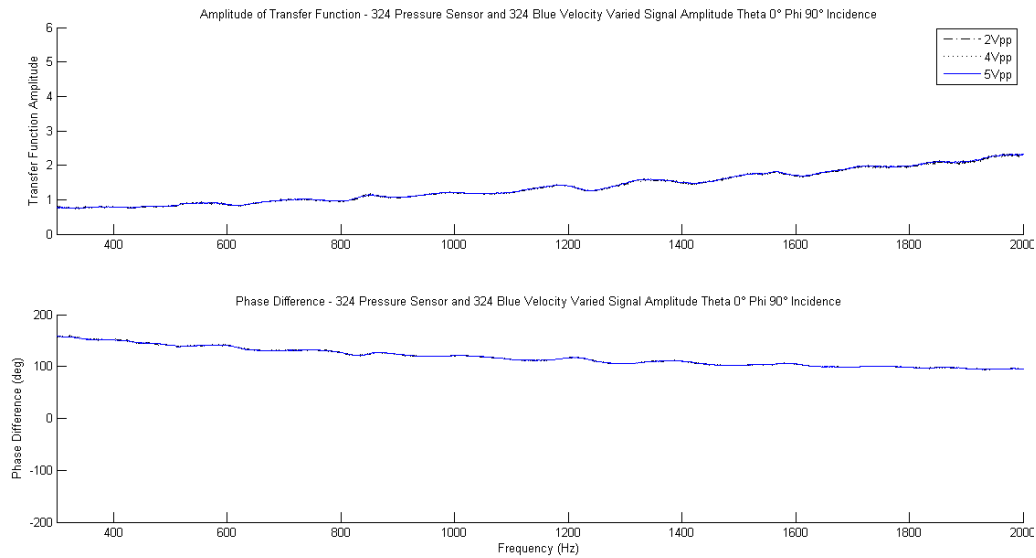


Figure 46. Sensor 324 Blue Element Transfer Function at Various Signal Amplitudes, Ignoring 1 Volt Signal

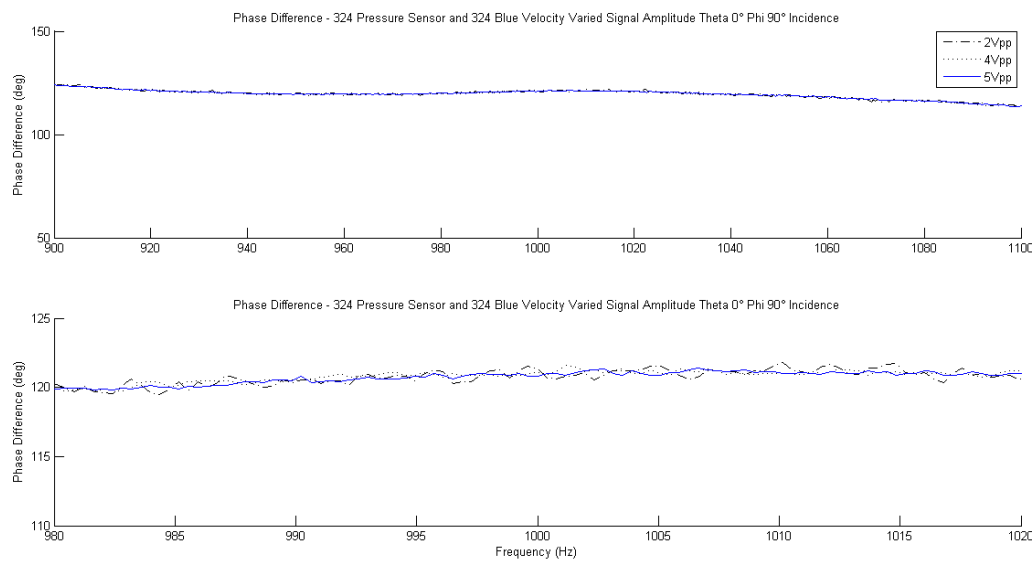


Figure 47. Sensor 324 Blue Element Transfer Function at Various Signal Amplitudes in Band of Interest, Ignoring 1 Volt Signal

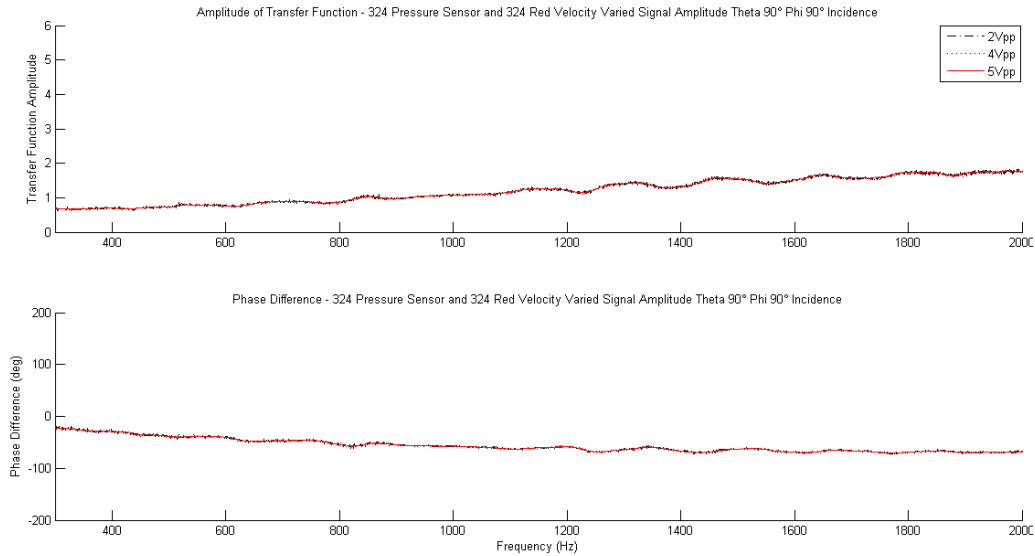


Figure 48. Sensor 324 Red Element Transfer Function at Various Signal Amplitudes, Ignoring 1 Volt Signal

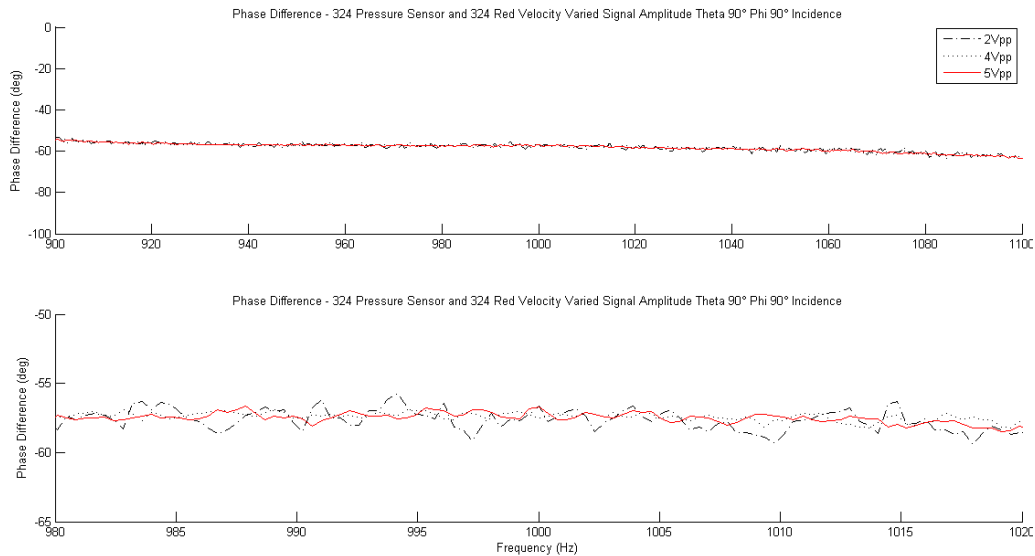


Figure 49. Sensor 324 Red Element Transfer Function at Various Signal Amplitudes in Band of Interest, Ignoring 1 Volt Signal

Discarding the 1Vpp result, the three remaining traces agreed well in both amplitude and phase, indicating the response of the velocity elements is not signal amplitude dependent. Even including the 1Vpp signal, the results agreed well. The lower coherence of the 1Vpp signal simply produced a transfer function with more noise, but the average trend line of the 1Vpp signal agreed

closely with the other three traces. For all practical purposes, given sufficient signal to noise ratio, the responses of the velocity elements are not amplitude dependent.

G. RESPONSE TO SIGNALS ARRIVING FROM DIFFERENT ASPECTS

Tests were also conducted maintaining the source signal amplitude as a constant and rotating the sensor to different polar angles in order to determine whether the phase of the velocity element responses were aspect dependent. The data was recorded using a signal amplitude of 5Vpp at six separate angles, the +MRA and -MRA and at 30° increments off the MRAs. These tests were also conducted on the blue and red elements separately. All signals produced excellent coherence with both elements, further confirming the independence of the phase response, even when off axis.

The extreme limits of the aspect test produced poor results, similar to the amplitude tests. As expected, the 60° off MRA signals produced distinct separation in amplitude response from the other signals. The amplitudes of the extreme off axis transfer functions were higher, consistent with a lower signal level and a larger correction factor. The MRA and 30° off axis traces agreed well on both the +MRA and -MRA traces. Figures 50–55 show the results of these tests.

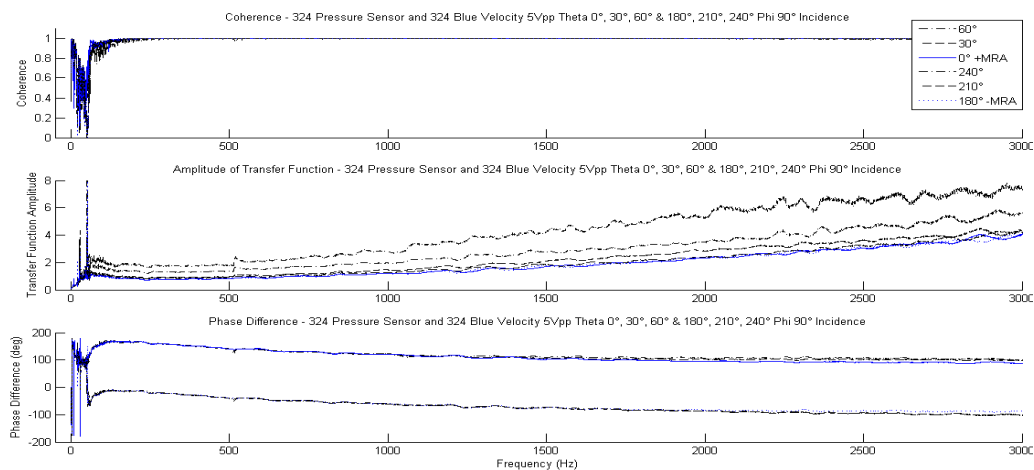


Figure 50. Sensor 324 Blue Element Transfer Function at Various Aspects

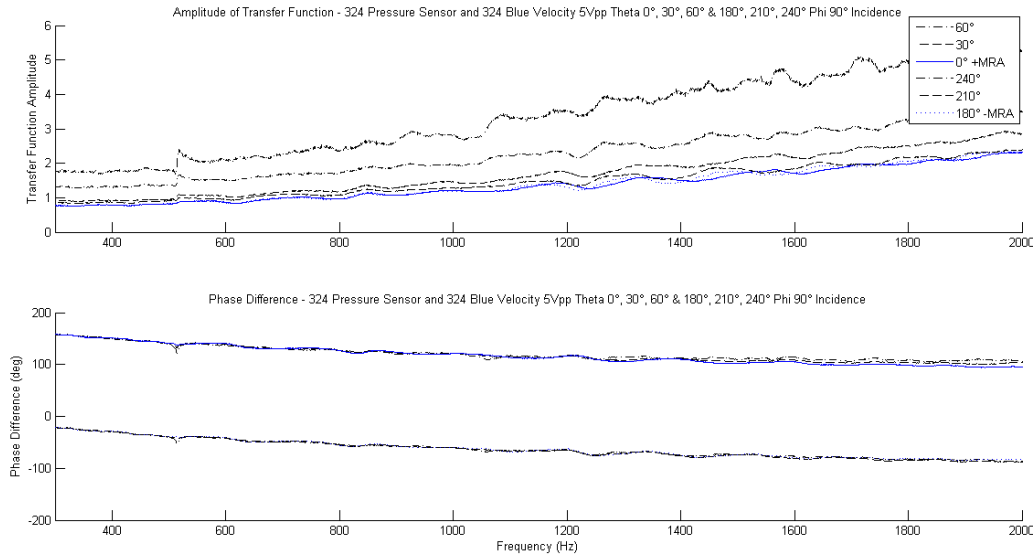


Figure 51. Sensor 324 Blue Element Transfer Function at Various Aspects in Band of Interest

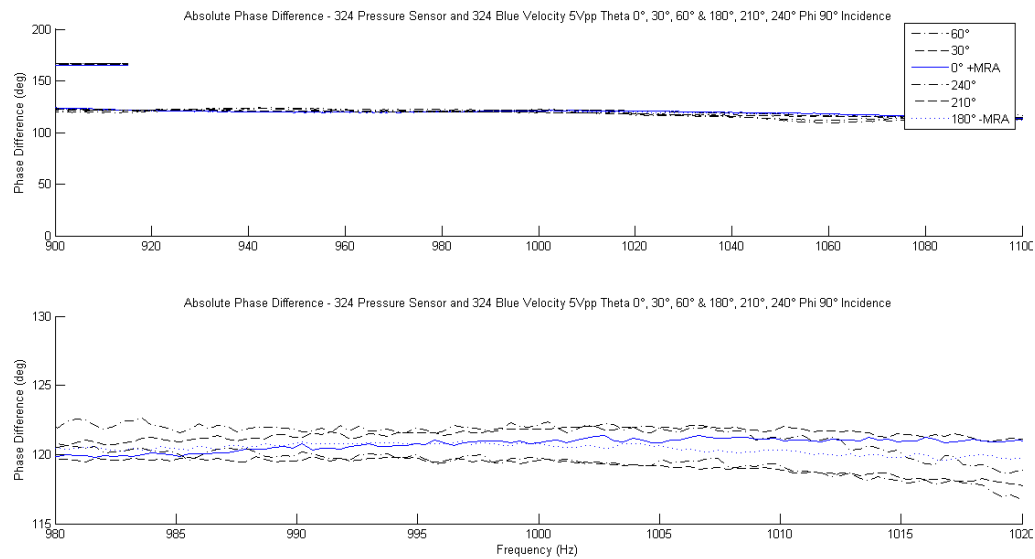


Figure 52. Sensor 324 Blue Element Transfer Function at Various Aspects Centered at 1kHz

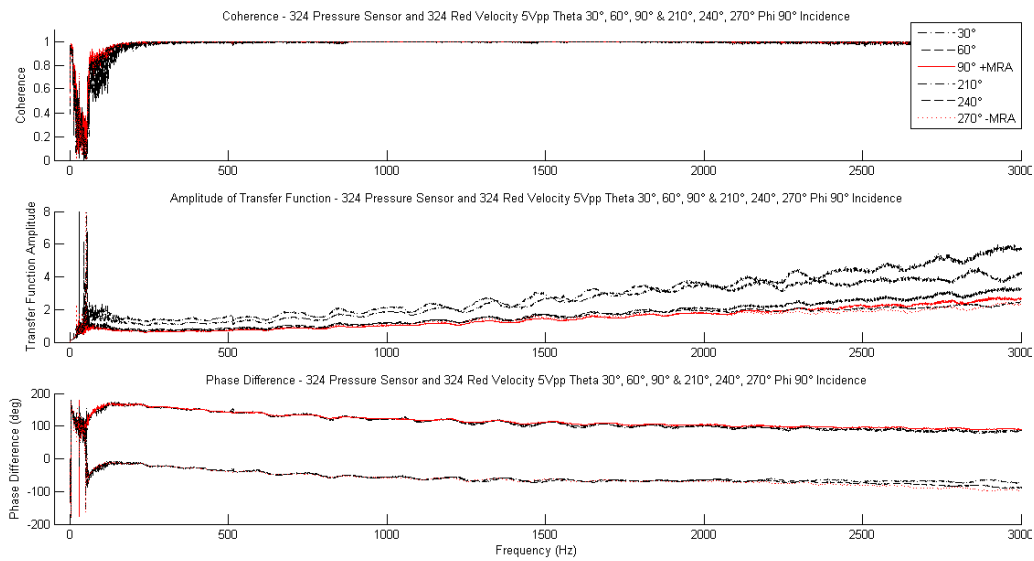


Figure 53. Sensor 324 Red Element Transfer Function at Various Aspects

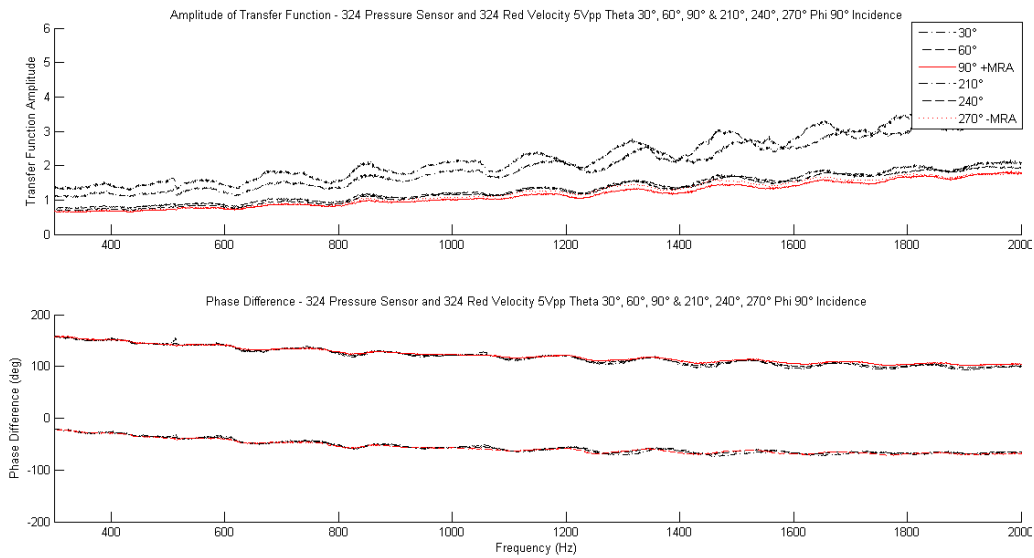


Figure 54. Sensor 324 Red Element Transfer Function at Various Aspects in Band of Interest

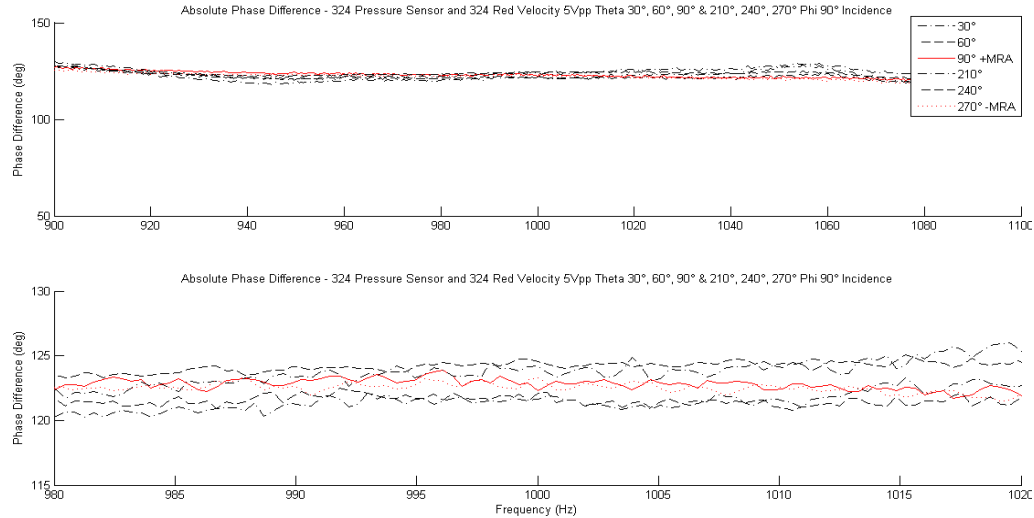


Figure 55. Sensor 324 Red Element Transfer Function at Various Aspects Centered at 1kHz

The phase response of all signals, including the extreme off axis signals, agreed well. There was slight divergence of a few degrees in phase observed mostly above 1.5kHz. As the frequency increases, the wavelength gets shorter and the effects of path length differences of the rotated element emerge. This is because the velocity elements are displaced from the center axis of the sensor by a few millimeters. As the sensor is rotated about its central axis, the velocity elements revolve about the central axis at a displaced radius. For example, a difference in path length along the MRA of 2mm equates to a maximum phase difference of 2 degrees at 1kHz. This is consistent with the magnitude of the phase shift observed in Figure 55. Given this small effect, the transfer functions were taken to be constant with respect to aspect angle for the remainder of the study.

H. DIFFERENCE BETWEEN PRESSURE ELEMENT TRANSFER FUNCTIONS OF THE RED AND GREEN MRA ORIENTATIONS

Two separate dataset series were collected to determine the Microflown pressure element to Aco transfer functions. One set was collected with the apparatus in the red element MRA orientation and the other collected with the apparatus in the green element MRA orientation. The blue element MRA

orientation was not used. This is because the blue element MRA lies along the z axis. The MRA of the blue sensors results in the linear axis of the array pointing directly toward the source. This orientation places the 323 sensor in front of the Aco and the 324 sensor behind the Aco. This maximizes the path length difference between the sensors and would alter both the true amplitude and phase calculations. This is also the reason the transfer functions for all of the velocity elements were calculated relative to their onboard pressure sensors, and then multiplied by the pressure element to Aco transfer function. It is simply physically impossible to place the apparatus in all three MRA orientations and not introduce path length differences between the sensors in the blue orientation.

The six resultant transfer functions were plotted against each other. All six signals showed excellent coherence. As before, the three transfer functions in each orientation, green and red, agreed with one another very well. The green and red amplitude traces agreed closely, confirming the near omnidirectionality of the pressure elements. The phase functions, however, diverged significantly from the 100Hz point and higher. Figures 56 and 57 show this comparison for both Microflown sensors.

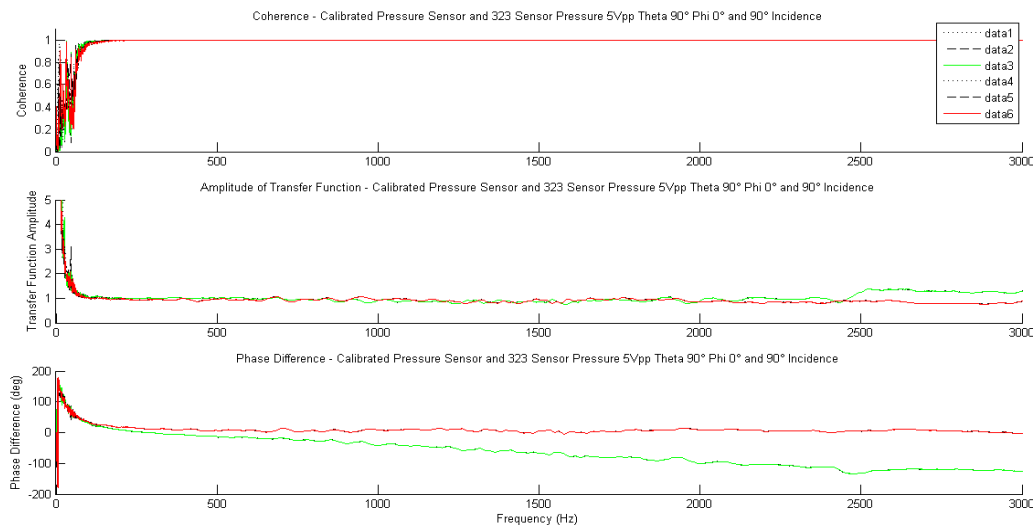


Figure 56. Sensor 323 Pressure Element Transfer Function at Green and Red Element MRAs

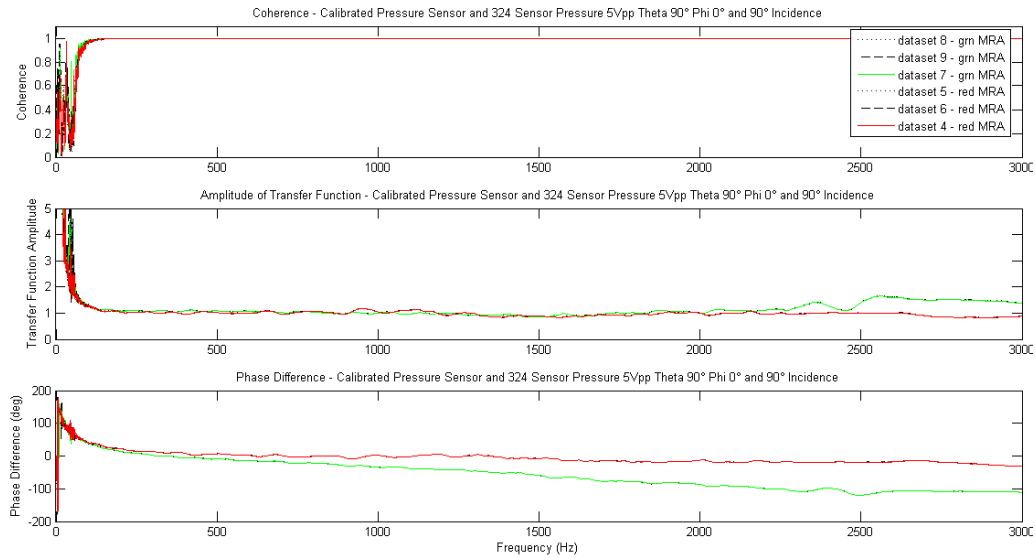


Figure 57. Sensor 324 Pressure Element Transfer Function at Green and Red Element MRAs

With the apparatus in the green MRA orientation, the sensors are pointed directly at the source. The signal wave is incident normal to the face of the velocity elements. With the apparatus in the red MRA orientation, the sensors are pointed down, with their central axes perpendicular to the source. The signal wave is incident parallel across the face of the velocity elements. It is possible there were some unaccounted for diffraction effects with the apparatus in the red orientation. The decision was made to use the pressure element transfer function calculated from the data taken with the apparatus in the green MRA orientation since the plan for the live data acquisition was to collect signals of aircraft flying overhead.

VII. BEAMFORMER

A. METHOD

The beamformer code breaks up the portions of the summing algorithm detailed in Chapter II into manageable pieces. Several steps of the sum, which are not dependent on the incoming signal, are pre-calculated to reduce computational overhead in the final beamformer routine. Considering the size of the data arrays involved, a computer with a minimum of 2GB of Random Access Memory (RAM) is required to produce results with sufficient angular resolution. Tests using maximum steering angle increments of $\pm 5^\circ$ in both polar θ and azimuthal ϕ angles can be processed on computers with 1GB of RAM. These tests can give indications of proper beamformer performance, but will not produce high-resolution results. Even with sufficient RAM, it is necessary to limit the bandwidth of the beamformed signal to avoid running out of memory. In practice a limit of one octave above and below 1kHz, from 500Hz to 2kHz matches well with the computational load and the design of the array. Reducing the sampling frequency can also reduce the size of the processed arrays without losing much resolution in the desired region.

Programming note: The MATLAB code makes maximum use of indexing and reshaping of matrices to perform multiplication of three-dimensional arrays in two dimensions. Multiplication of two-dimensional arrays, then reshaping into three dimensions, is significantly faster than conducting the multiplication across three dimensions. The use of the *repmat* function is specifically avoided. The *repmat* function is actually a script and not a built in function, therefore it runs significantly slower and requires a larger percentage of computational resources. An effort was made to comment the code to sufficiently explain the size of the arrays at various points in the beamforming routine and to ensure the axes dimensions remain appropriate relative to one another.

The beamformer must have the proper parameters entered in the setup area, which correspond to the format of the binary data file and frequencies and steer angles of interest. The transfer functions must be available in a .mat file. This file is automatically produced by the separate transfer function averaging program, and the data file location must also be entered.

B. RUNNING THE BEAMFORMER

Once started, the beamformer initializes and loads the transfer functions. It calculates a matrix of the various steer angles, the steer angle direction cosines and the element unit vectors. The next step is to compute the dot product $\hat{u}_s \cdot \hat{u}_{emn}$. All of the element dot products are the same for identical elements across all sensors. The array is constructed so that the MRAs of all similar velocity elements on separate vector sensors, e.g., 323 blue and 324 blue, are aligned. Thus, the unit vectors for the same elements on different sensors are identical. Since the unit vectors are equal, their dot products with the steer angle matrix are also equal. This reduces computational and memory overhead a great deal. So long as the MRAs of individual sensors are aligned, there is only a need to calculate four dot products, regardless of the number of sensors in the array. In this three-sensor, twelve-channel array, memory overhead for the dot product matrix is reduced 67%, from a size of 12x181x181 to 4x181x181. Additional sensors could be added with no additional memory devoted to this array.

Since the pressure elements and Aco microphone are omnidirectional, all bins of the dot product corresponding to the pressure elements are given a value of “1”. Thus, the contributions of the pressure elements to the beamformer sum depend only on the amplitude and phase of the corrected signals, as computed in a standard linear summation of omnidirectional elements. The dot product matrix is then vectorized from a three-dimensional 4x181x181 matrix into a 4x(181*181) two-dimensional array. This is done to speed processing in the beamformer routine as discussed in Subsection A above. The final step prior to

entering the actual beamformer routine is to create a reduced length k vector with relevant values of the frequency bandwidth of interest. This truncates the signals prior to summing in order to reduce processing overhead. The remainder of the signals are discarded to improve performance and reduce memory load after they are multiplied by the transfer functions.

The signals are read in from the binary file and a Hanning window is applied to the time series data to reduce side lobes. The windowed signals are Fourier transformed into the frequency domain, the redundant half of the Fourier transform is discarded, and the now frequency domain signals are multiplied by their respective transfer functions to “correct” the signal. At this point all signals are now referenced to the center element, the Aco microphone in this case. The corrected signals are truncated to the bandwidth of interest before summing in the beamformer to save processing resources.

$$\begin{aligned}
 x_{mn}(t) &\Rightarrow x_{mn}(t)W_{mn}(t) \Rightarrow xw_{mn}(t) \\
 xw_{mn}(t) &\Rightarrow FT \Rightarrow Xw_{mn}(k) \\
 Xw_{mn}(k)H_{mn}(k) &\Rightarrow Xwc_{mn}(k) \\
 Xwc_{mn}(k) &< truncate > \Rightarrow Xwc_{mn}(k_{BW})
 \end{aligned}$$

In order to further reduce memory overhead, the α weighting term (Eq. 2.12) constants are pre-calculated prior to entering the beamformer, and the specific sensor weighting term is calculated for each sensor just prior to summing its responses. The weighting term is identical for all four elements of a particular sensor, so it only needs to be calculated once for each sensor then applied to all four elements. The beamformer routine multiplies each of the truncated element signals by the appropriate element dot product and the sensor α weight and sums the results. Since the Aco, or any other pressure only elements added to the array, have no velocity sensors, the dummy channels are not processed in order to speed computation. The resultant sum is a two dimensional matrix with dimensions $(m * n) \times p$, where m is the length of the θ_s vector, n is the length of the ϕ_s vector and p is the length of the k_{BW} bandwidth vector. The amplitudes of all signals in the bandwidth of interest are summed into a single

value across the k_{BW} dimension, which collapses the matrix into a $(m * n) \times 1$ vector. The vector is then reshaped into a two dimensional array with size $m \times n$. This actually represents a three dimensional matrix of size $\theta_s \times \phi_s$ with the summed signal amplitude values residing in the singleton dimension, an index of steer angle combinations and total responses in those directions.

$$Xcw_{mn}(k_{BW})(\hat{u}_s(\theta, \phi) \cdot \hat{u}_{emn}(\theta, \phi))\alpha(\theta, k_{BW}) \Rightarrow S(\theta, \phi, k_{BW})$$

All the two-dimensional plots use *imagesc* to show the beamformer output in all possible directions. The three-dimensional plots have their axes labeled to indicate degree of view rotation. Figure 58 shows the theoretical beam pattern of an N element linear array of omnidirectional elements for reference to the experimental beamformer output. Figures 59-62, show the beampattern of the experimental array for familiarization purposes. The datastreams were processed by the beamformer as both a standard omnidirectional array, by disabling the velocity element inputs, and as a hybrid array using the all sensor elements. Any comparison images below were processed using identical data segment(s) for each orientation.

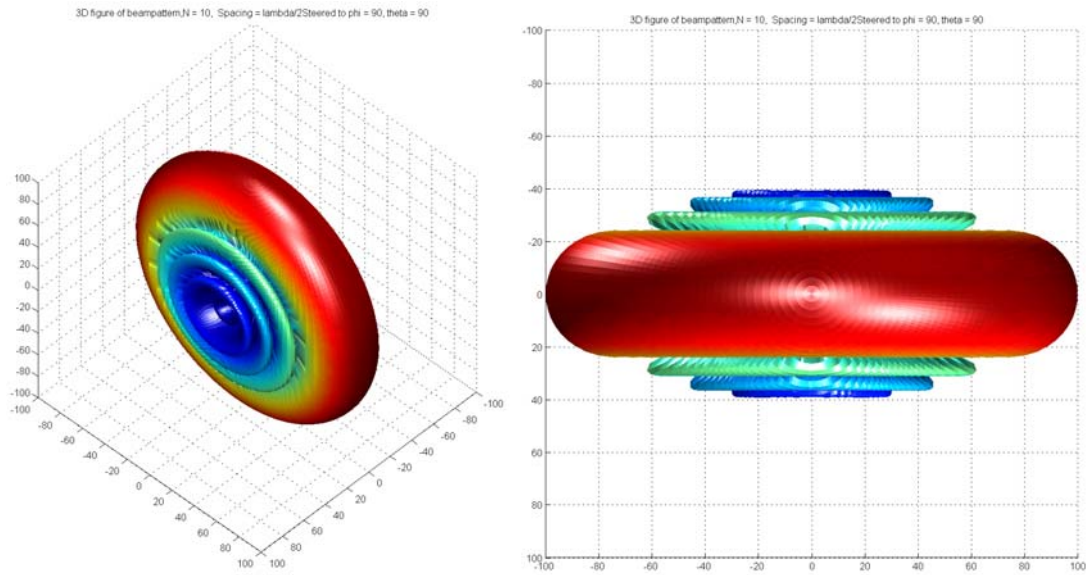


Figure 58. Theoretical 3D Beampattern For an N Element Omnidirectional Linear Array

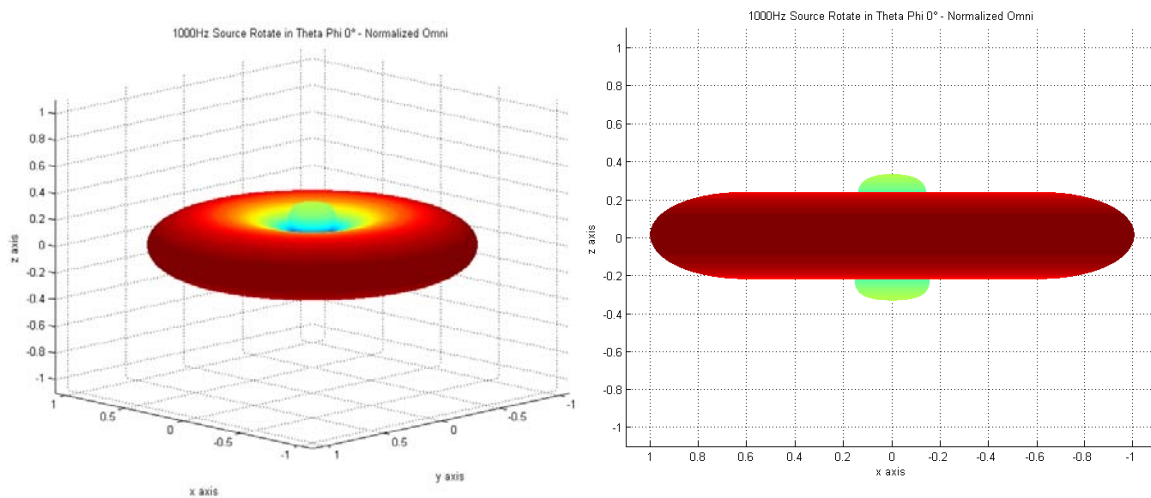


Figure 59. Normalized 3D Beampatterns of the Omni Array With a 1kHz Source At $\theta = 90^\circ$ and $\phi = 0^\circ$ Showing Bearing Ambiguity

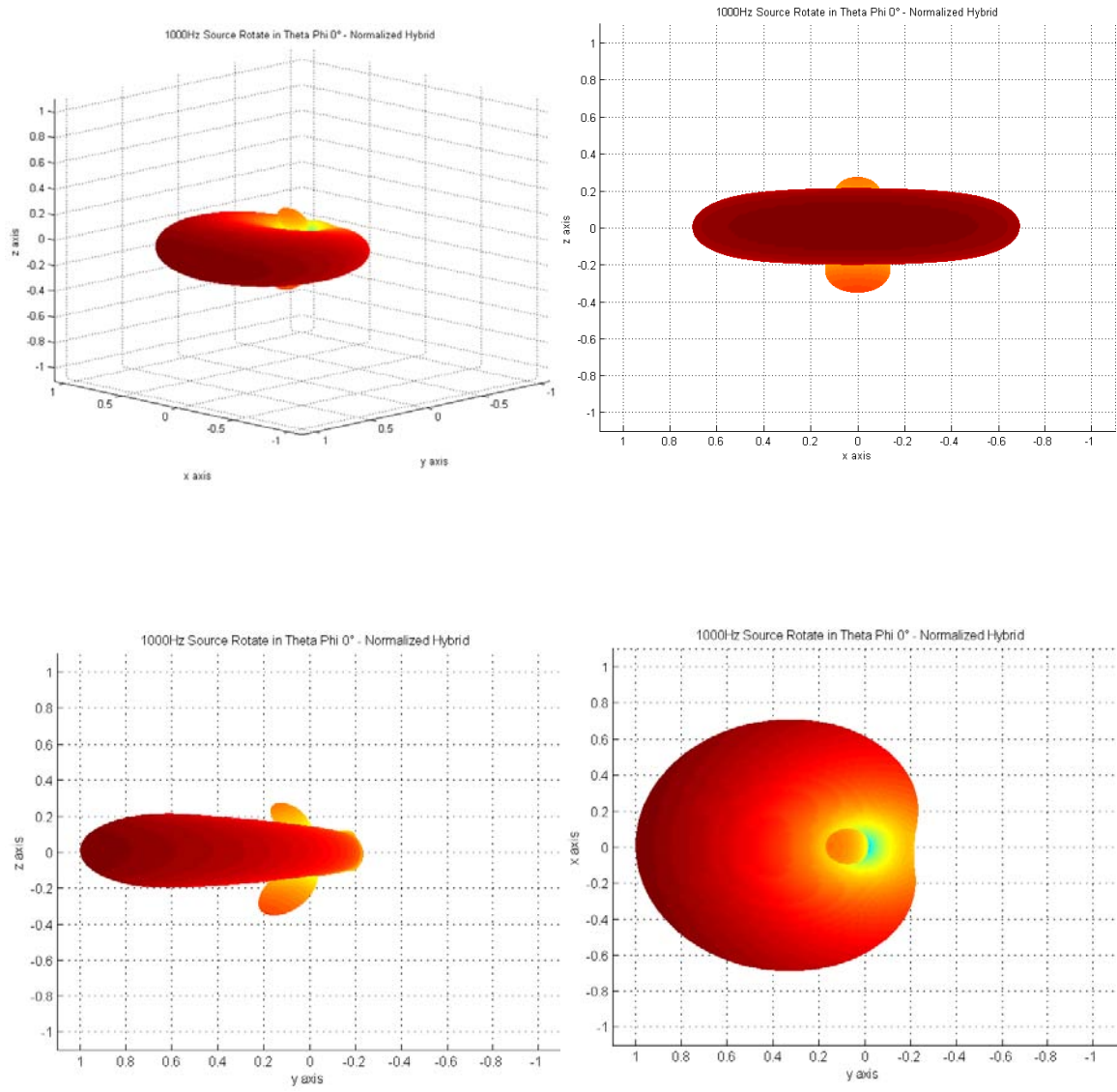


Figure 60. Normalized 3D Beampatterns of the Hybrid Array With a 1kHz Source At $\theta = 90^\circ$ and $\phi = 0^\circ$ Showing Beam Focusing

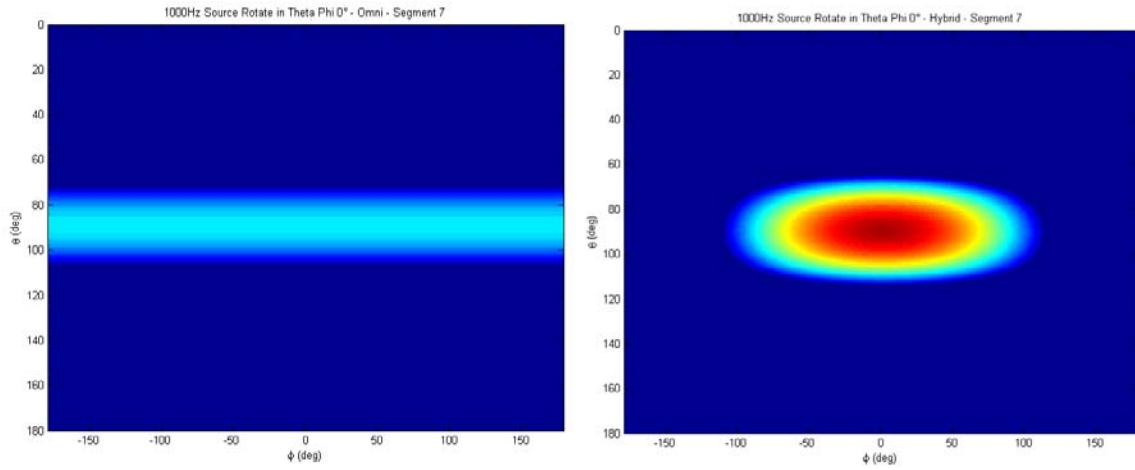


Figure 61. Beampattern of the Omni and Hybrid Arrays With a Source of 1kHz At $\theta = 90^\circ$ and $\phi = 0^\circ$ on the Same Intensity Scale

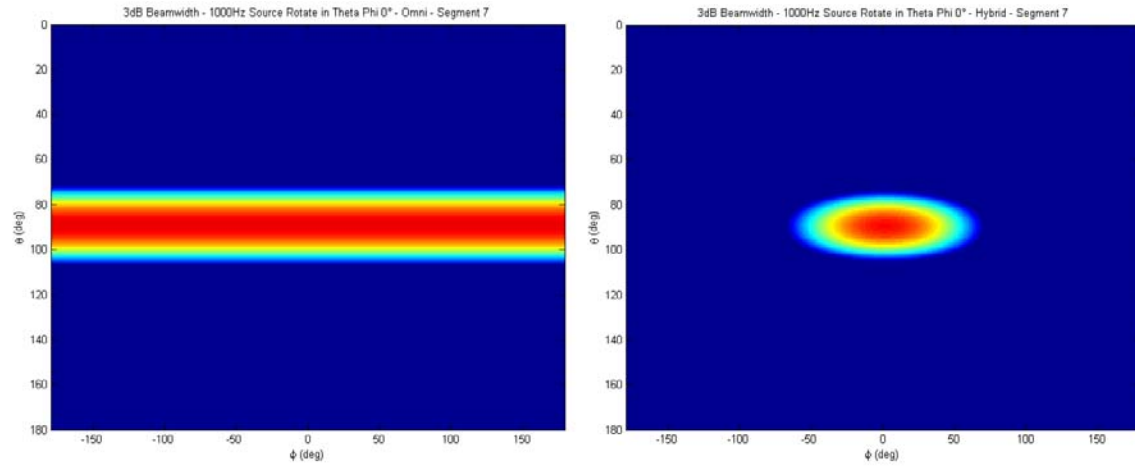


Figure 62. 3dB Beamwidth Beampattern of the Omni and Hybrid Arrays With a Source of 1kHz at $\theta = 90^\circ$ and $\phi = 0^\circ$ on Respective Intensity Scales

C. STATIC TEST WITH SINGLE SOURCE

Several static tests were conducted with the array in known orientations and the source at a known polar angle relative to the origin of the array. Signals of 300Hz, 500Hz, 1kHz and 1.5kHz were used. The datastreams were processed by the beamformer as both a standard omnidirectional array, by disabling the velocity element inputs, and as a hybrid array using the all sensor elements. Any comparison images below were processed using identical data segments for each orientation. The data was plotted using *imagesc* in θ and ϕ space, and as three dimensional beam patterns using *surf*. The beamformer determined the correct location of the source relative to the array with a high degree of accuracy. Figure 63 shows the results as the source polar position was changed. Orientation of the array did not appear to have a large effect on the bearing accuracy of the beam pattern. When a source was placed on the positive or negative z axes in the "end fire" region, the two-dimensional plots showed beam broadening due to the nature of the flat projection of a curved surface. The three-dimensional plots in these cases showed similar beamwidths to the broadside plots. When a source was placed in the broadside region, the beam patterns of both plot types were similar.

Note the last two omni array images of the MRA plots in the second set of images, Figure 64. They are nearly indistinguishable from one another. Yet their corresponding hybrid plots clearly show the source in a different azimuthal position. The main lobe of the omni plot when the source travels over the pole of the z axis, first on the left in Figure 64, appears to be 180° out of position. This only occurs when going over the pole and only on the omni plots. The reason is unknown. Due to mass of the apparatus and the non-rigid connection, some drooping occurred when in certain orientations. The actual source aspect in the beam patterns may be slightly different than the recorded orientation. This is readily apparent in Figure 64 where the true polar aspect is 80° vice the recorded polar aspect of 90° .

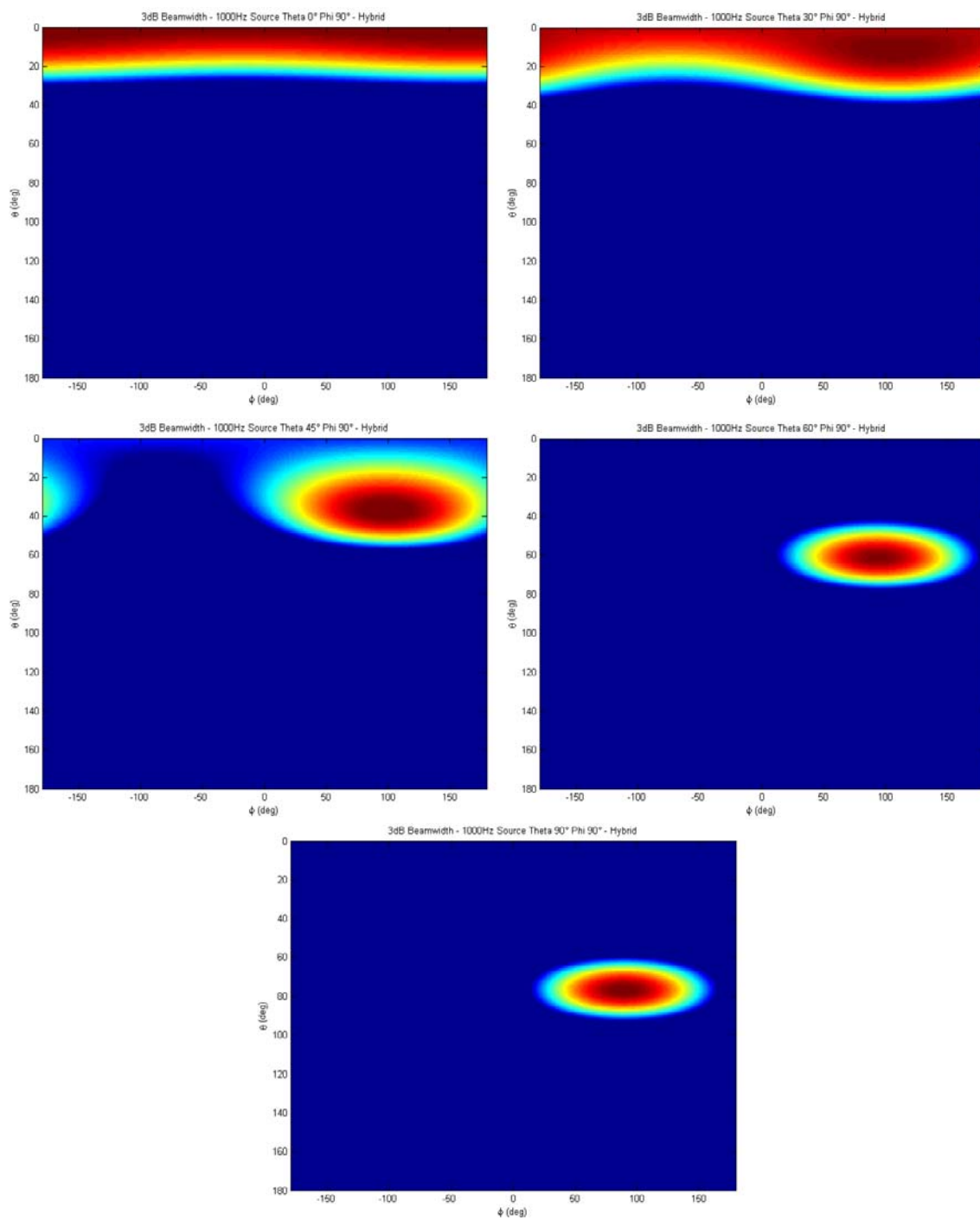


Figure 63. Comparison of Hybrid Array Beam Patterns for a 1kHz Signal Placed at Various Relative Aspects in θ Along the ϕ Planar Horizon

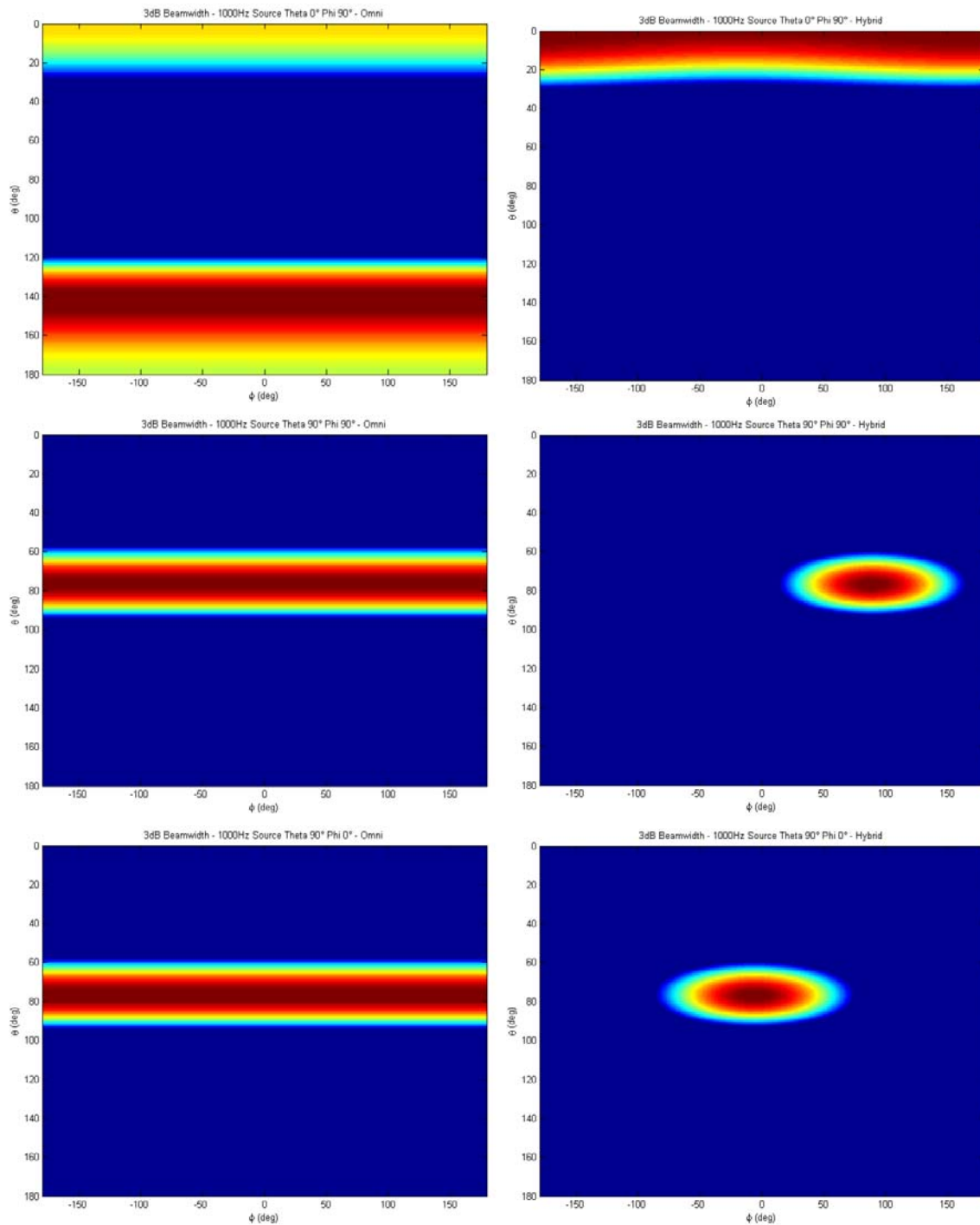


Figure 64. Comparison of Omni and Hybrid Array Beam Patterns for a 1kHz Signal Placed at The Three Primary Velocity Element MRAs.

D. DYNAMIC TEST WITH SINGLE SOURCE

Results that are more interesting were achieved with the array in motion. Several dynamic tests were conducted with the array in several known orientations relative to the axis of rotation. A source was placed a fixed distance from the array and the rotator mechanism energized. This simulated a stationary array with a target source moving circularly around the array about a single axis.

The images on the left, Figure 65, show the response of the omnidirectional array, those on the right show the directional hybrid array. The source began at the $\theta = 90^\circ$ $\phi = 90^\circ$ position, the array was rotated about the x axis, simulating the source passing across the z axis and around to the other side. The top and bottom images on the left are nearly indistinguishable from one another, there is no way to determine relative source aspect. It is important to note this two dimensional image represents a three dimensional ring or band around the long z axis of the array. There is no way to resolve the bearing ambiguity of the omni array without further data and time. The images on the right side clearly show instantaneous position of the source relative to the array, in three dimensions.

The middle pictures of both arrays look very similar. This is a result of the inter-element spacing and array length perfectly matching the wavelength of the 1kHz source signal and the data spreading of a spherical object onto a two dimensional plot. The two Microflown pressure sensors are detecting peak signals exactly one wavelength apart, while the Aco sensor in the middle, aligned π radians from the Microflown sensors, is in a trough. Note the lack of a lower lobe on the hybrid array plot indicating directionality even at end fire. All angles in ϕ coincide with the same point in space when the source is near or at $\theta = 0^\circ$, this produces the apparent broadening of the beam lobe on the plot. For the hybrid array, this is a transient condition that quickly resolves once the source passes over the pole. This provided direction of movement information during the transition. The omni array provided no such indication. Figure 66 shows the

source crossing over the pole and the rapid resolution of aspect by the hybrid array, including positional indication immediately before and after passing over the pole.

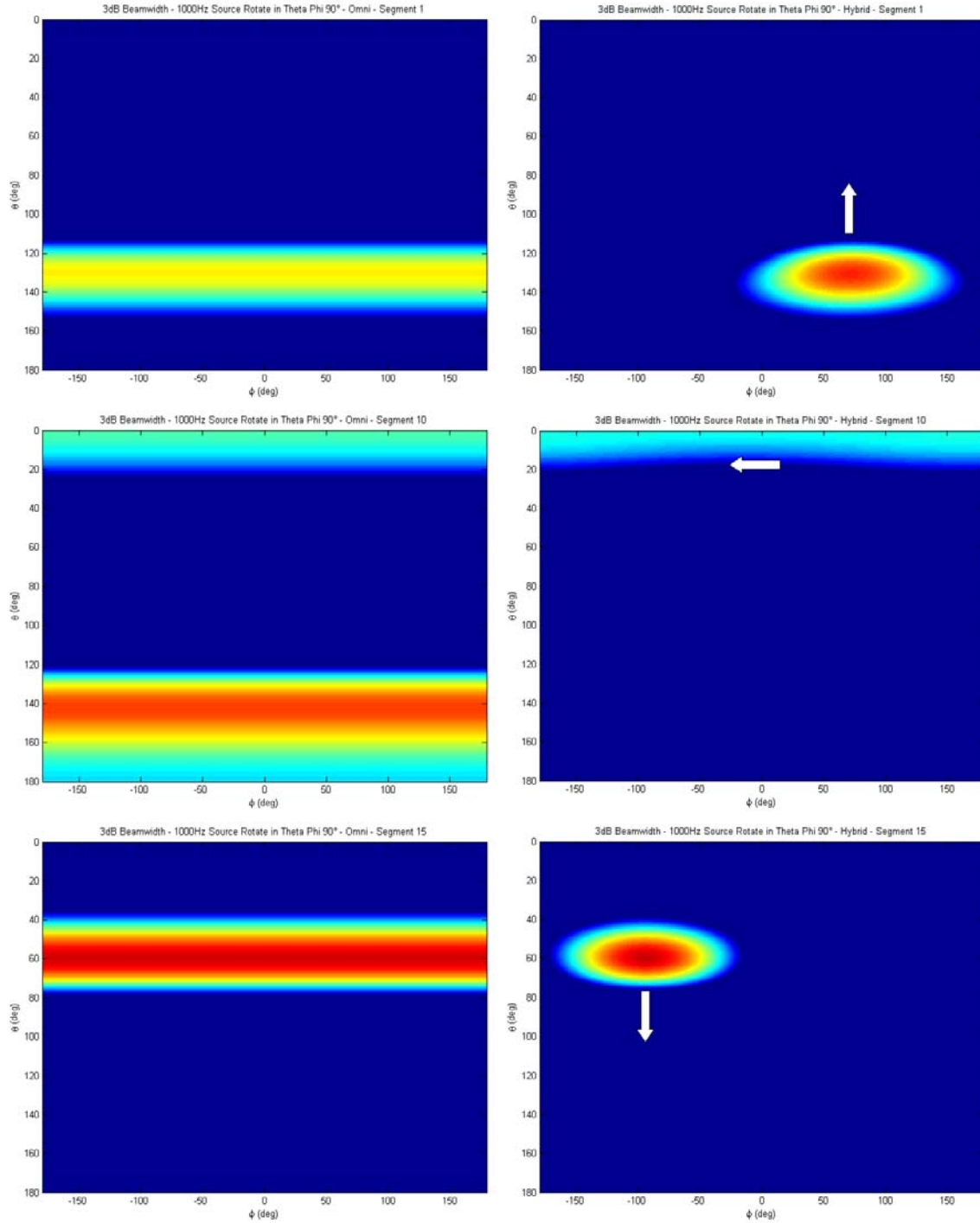


Figure 65. Source Tracking of Omni Array vs. Hybrid Array Using a 1kHz Source Moving About the x Axis. (Arrow Denotes Direction of Travel)

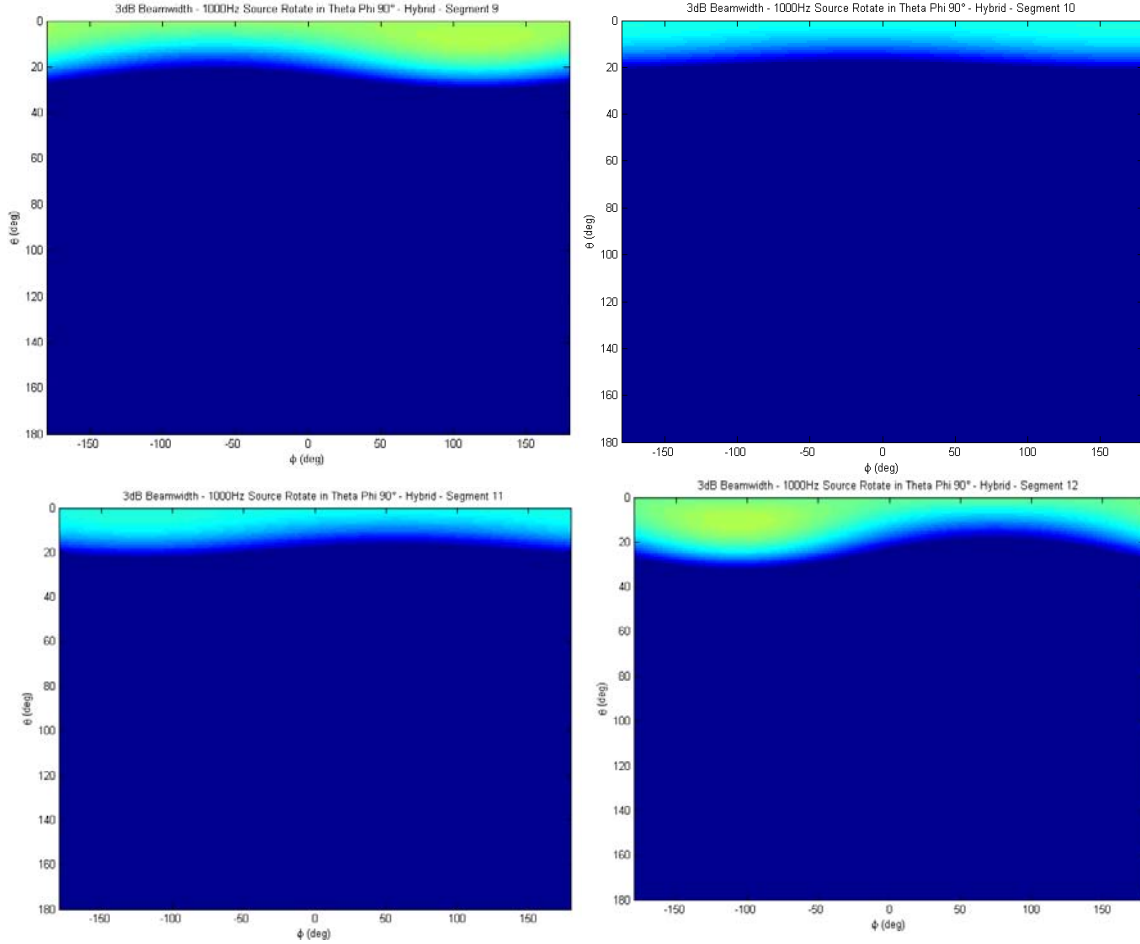


Figure 66. Source Tracking of Hybrid Array Showing Rapid Bearing Resolution Over the Pole (Elapsed Time ~10 sec)

E. DIRECTIONALITY / BEARING AMBIGUITY

The scales of the second sets of plots were relative to their respective arrays. The plots use the maximum response of the particular array as the upper limit and 3dB below this as the lower limit. This determines the 3dB beam width and directionality of the two arrays. The omni array plots displayed the traditional beam pattern with circular symmetry about the z axis of the array, the traditional broadside or “beamfire” pattern. As expected, it was impossible to determine a bearing to the target in the ϕ axis using the omni array, and there was bearing symmetry causing the traditional bearing ambiguity. The omni array could track movement along the z axis, but there was no way to determine on which side of

the array the source was located. The hybrid array plots showed very good directionality. The beam patterns were distinctly oval shaped, rather than the rectangular bands of the omni array. Right or left aspect could readily be determined as evident in Figure 67.

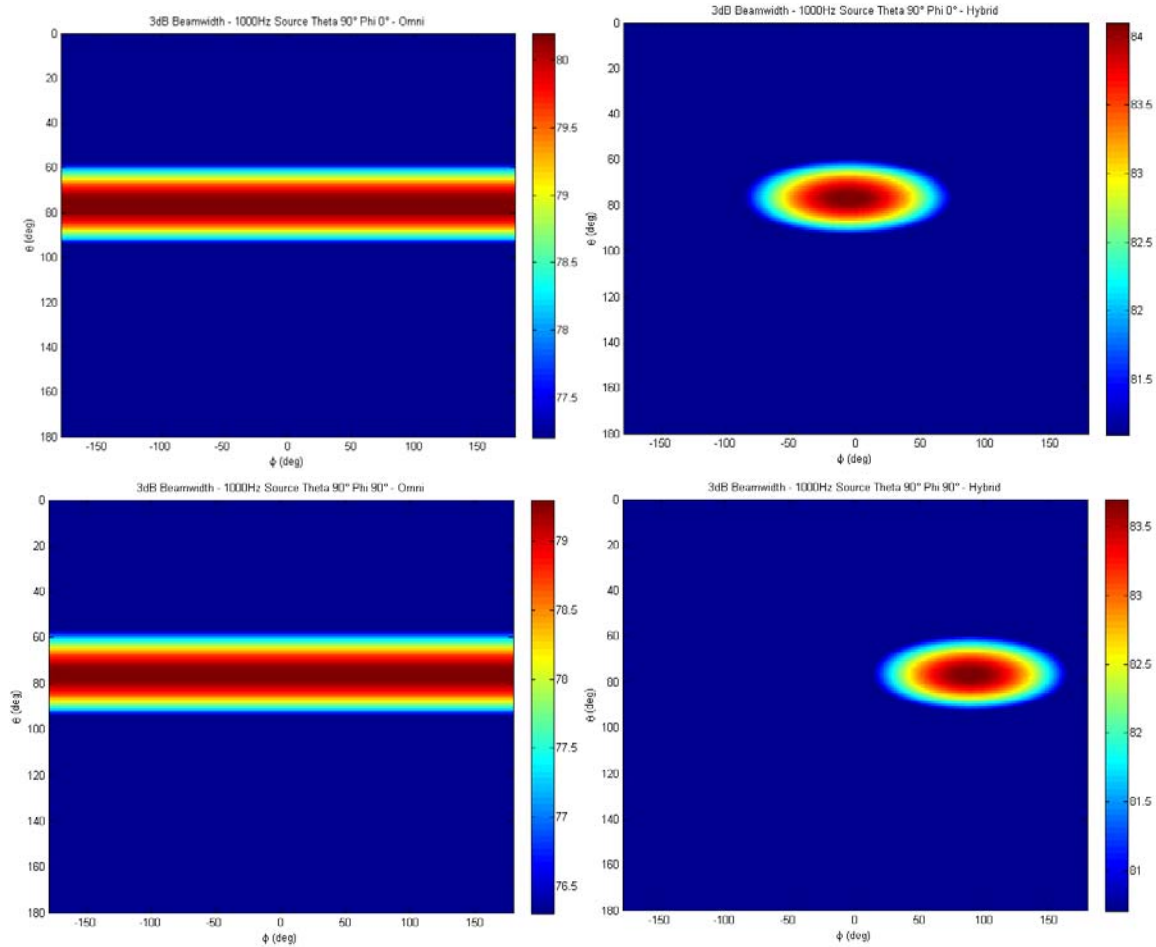


Figure 67. 3dB Beamwidth Differences of Both Arrays from a 1kHz Source Plotted with Color Scales Relative to Each Array

F. BEARING ACCURACY

Bearing accuracy of the hybrid array was adequate. Two methods were used to determine coarse and fine resolution, 3dB and 1dB half-beamwidths. A measure of the 3dB and 1dB beamwidths in degrees were made with several sets of data. For example, the nominal bearing accuracy of the hybrid array at the design frequency with the source at $\theta = 90^\circ$ was, $\pm 12^\circ$ in θ and $\pm 35^\circ$ in

ϕ for the 3dB beam width, and $\pm 3.5^\circ$ in θ and $\pm 12^\circ$ in ϕ for the 1dB beamwidth. Figures 68 and 69 show an example. Since there is no spatial separation of the sensors in the azimuthal plane, the beamwidth is wider on this axis than in the polar plane. A planar array with spatial separation in two directions of the elements would likely reduce the larger width of the beam to a value closer to the narrow width and produce a more circular beam.

Tracking was readily possible using the hybrid array. The rotator mechanism has an angular translation rate of approximately $6^\circ/\text{sec}$. Each chunk of data processed contained 65,536 (2^{16}) datapoints. At a sampling frequency of 25.6kHz, this translates into 2.56s per frame, or approximately 15° . This was readily monitored on the plot movies.

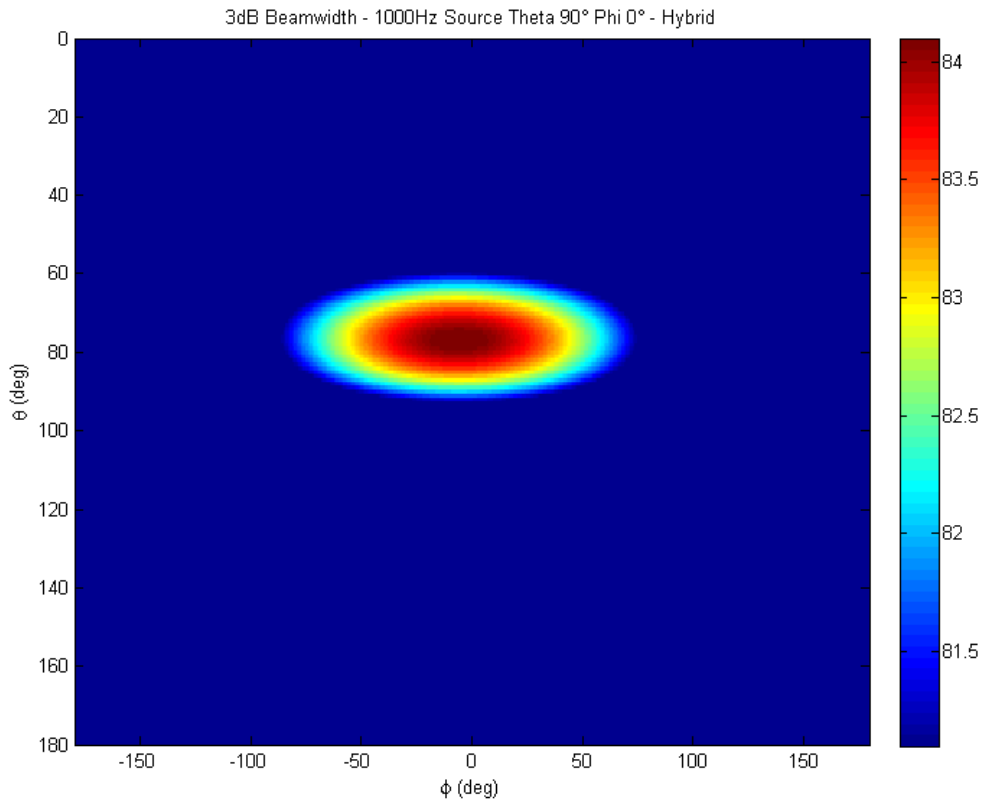


Figure 68. 3dB Beampattern of The Hybrid Array for an Overhead 1kHz Signal

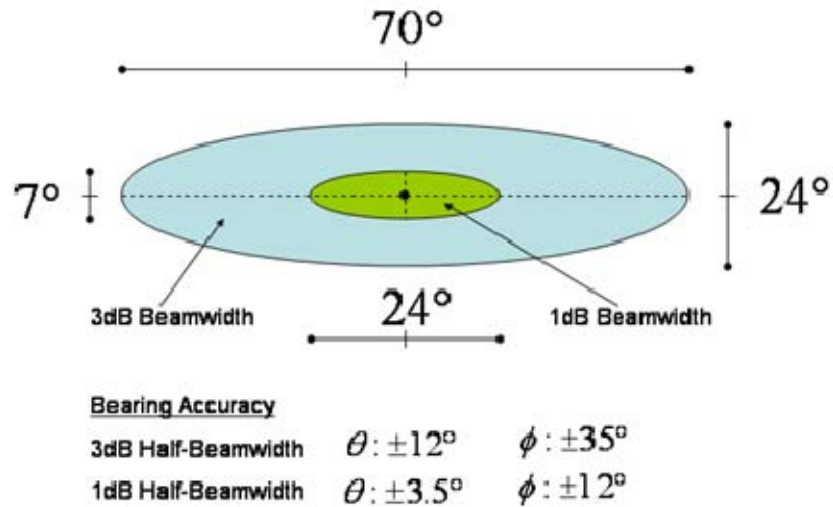


Figure 69. Bearing Accuracy of The Hybrid Array for a 1kHz Signal, Using Half-Beamwidth Method

G. ALTERNATE FREQUENCY PERFORMANCE

Beamformer performance was also tested with source frequencies of 1.5kHz, 500Hz and 300Hz. The absolute lower limit for this array is 300Hz. Beyond that limit, the transfer functions lose coherence drastically. The same general beam patterns and directionality of the hybrid array versus the omni array were observed. As shown in Figure 70, the performance was nearly identical using the 1.5kHz signal, with a slight narrowing of the beam pattern in the θ direction, due to the shorter wavelength. Since the array was optimized for 1kHz, using an inter-element spacing of $\lambda / 2 = 0.170m$, sidelobes started to appear with the 1.5kHz signal. This was observed in Figure 71 as multiple hot spots in the beam patterns. These are not visible in the 3dB plots. Higher frequencies would cause even larger amplitude sidelobes, eventually showing on the 3dB plots. The longer wavelength of the lower frequency signals combined with the fixed aperture size optimized for 1kHz caused beam broadening with the 500Hz and 300Hz signals which can be seen in Figures 72 and 73. Figure 74 shows a comparison of array design frequency to source frequencies and the effect on beam patterns.

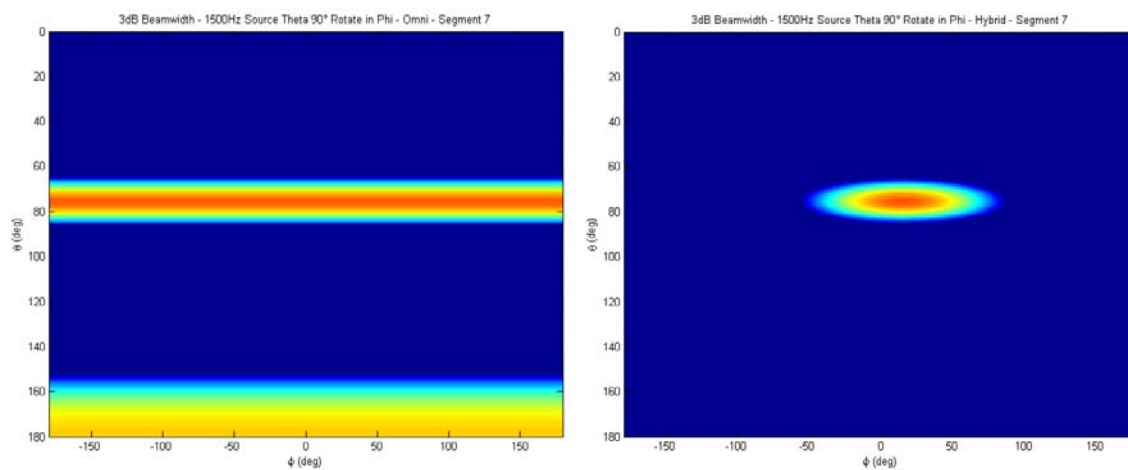


Figure 70. Beampatterns of the Arrays with a 1.5kHz Source

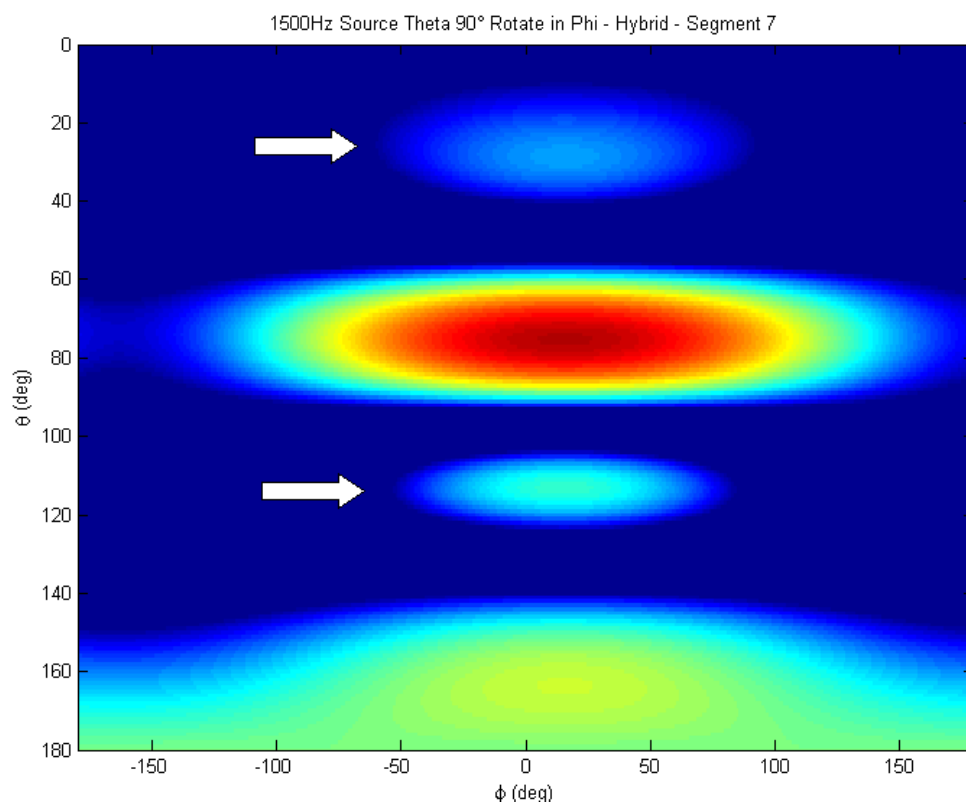


Figure 71. Beampattern with a 1.5kHz Source Showing Sidelobes

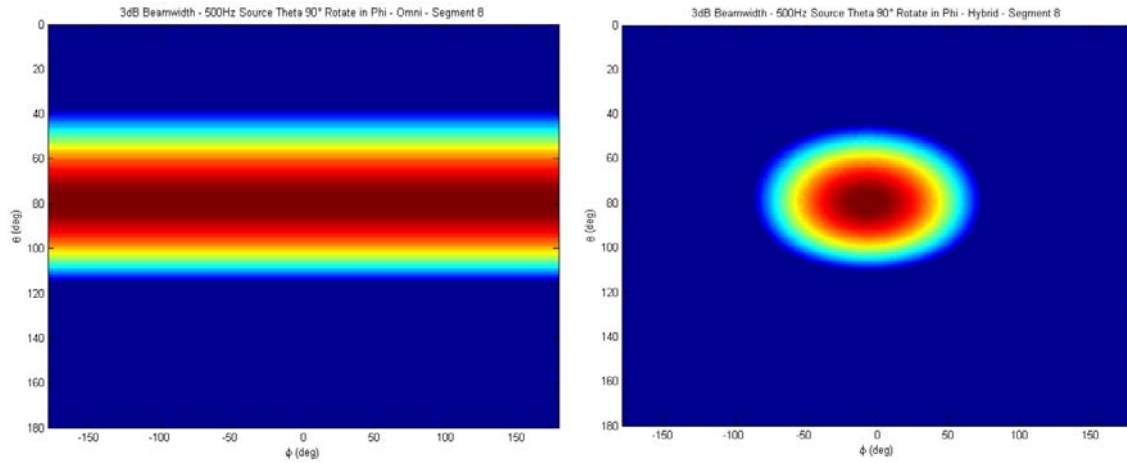


Figure 72. Beampatterns of the Arrays with a 500Hz Source

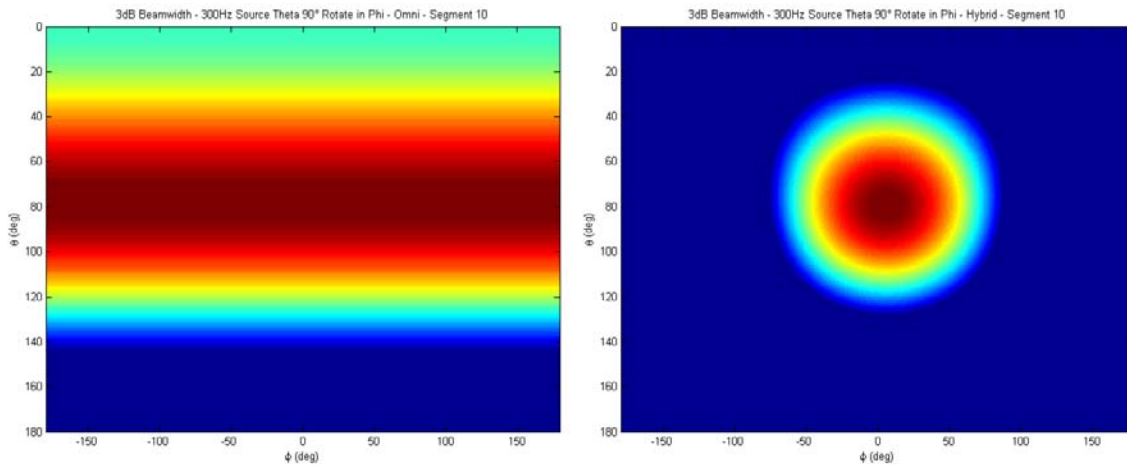


Figure 73. Beampatterns of the Arrays with a 300Hz Source

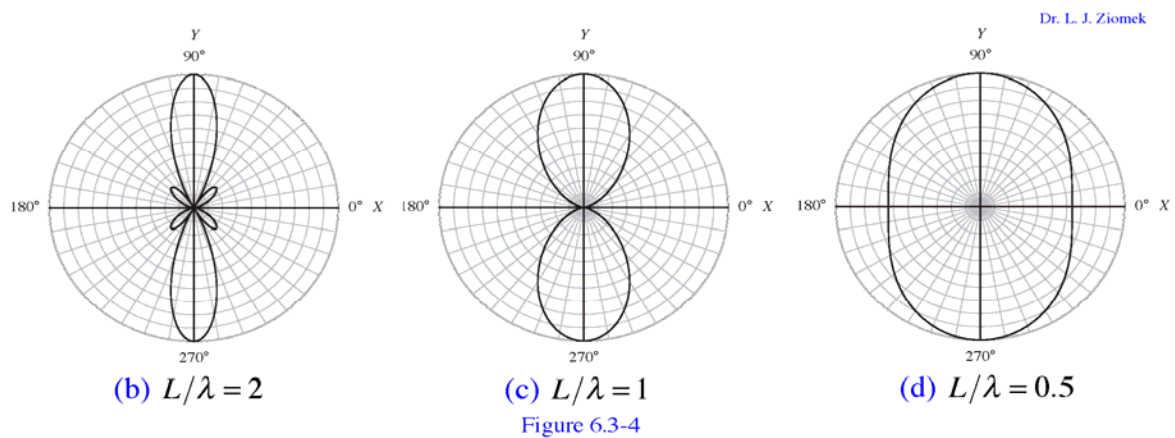


Figure 74. Sidelobes and Beamwidth Broadening with Various Array Length/Frequency Ratios of Continuous Line Arrays (From: Ziomek, 1994)

H. MULTIPLE SOURCE DISCRIMINATION

Several tests in multiple configurations were conducted with two sources. The sources were placed at two different angular separations. First at equal aspects on the polar plane separated 60° in the azimuthal plane, and then separated 20° in the polar plane and at equal aspects in the azimuthal plane. The loudspeakers were driven with sources of identical frequency as well as separate frequencies in both arrangements. The array could readily and accurately discriminate between the two sources when they had different frequencies, even with the narrow spatial separation of the second set up. Note the location of the 1kHz source in the various plots is the same in each grouping (wide and narrow). The speakers were not moved during the groups of tests, only the driving frequencies were adjusted.

There were no real indications of source separation when the sources had the same frequency, as in Figure 75. It generally located the source with the higher amplitude corresponding to alignment with an element MRA, but was blind to the other source. When in motion the array plots displayed hot spots for the individual sources when processed at their respective frequencies. An attempt was made to expand the bandwidth of the processor to capture both source center frequencies and process the data simultaneously. This produced unsatisfactory results.

The best results were obtained when processing the data as if they were independent single sources. The frequency of the first source was used as the center bandwidth frequency on the first processing run, and the source frequency of the second source was used for the second processing run, then the results were viewed separately. This is shown in Figures 76 and 77. In practice, for a tracking device where the source frequency was unknown, the frequency could be swept through the spectrum of interest, processed individually, and displayed as separate images.

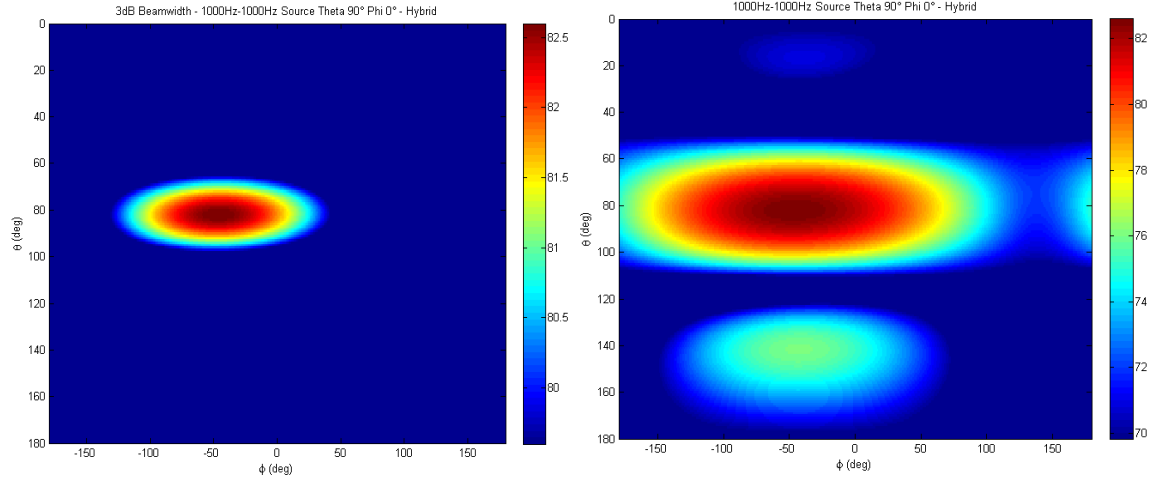


Figure 75. Beampatterns of the Hybrid Array with Two Sources at the Same Frequency and a Spatial Separation of 60° in θ

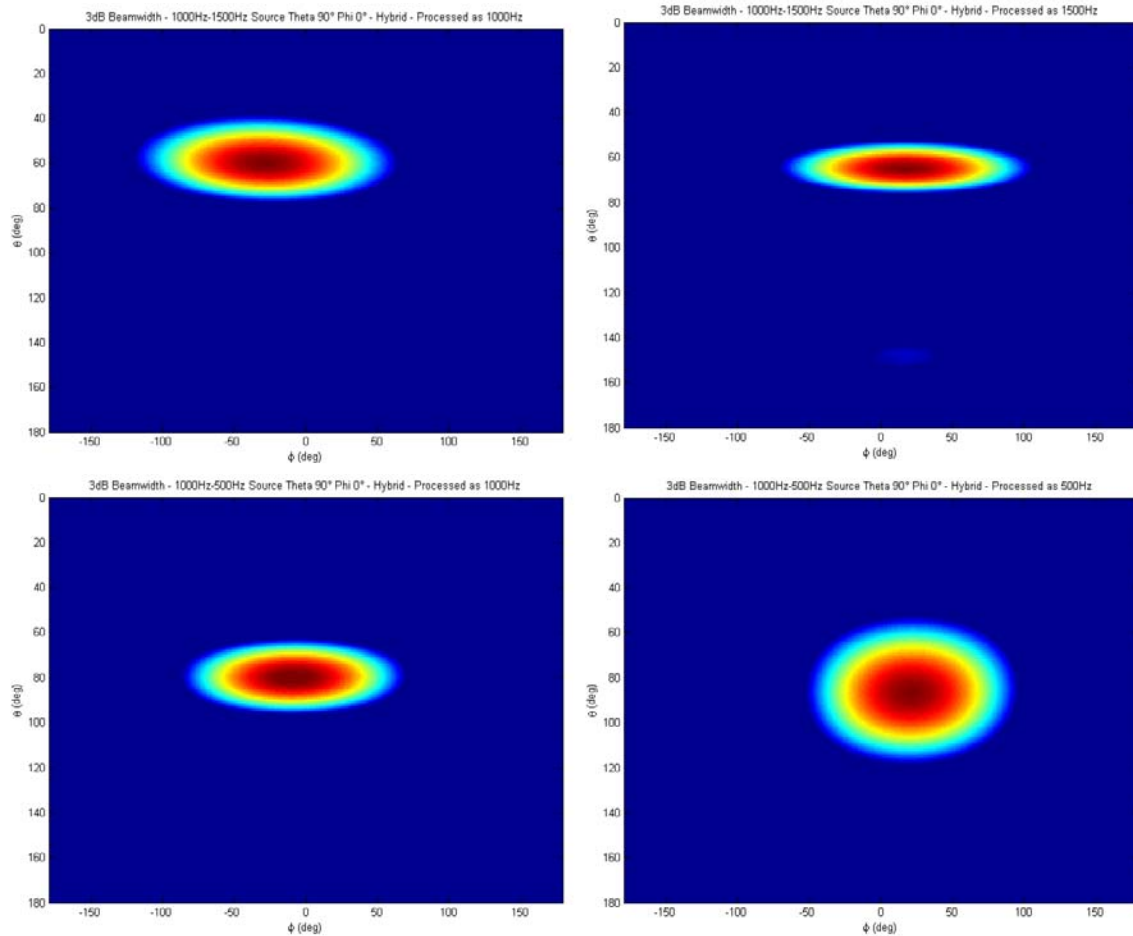


Figure 76. Beampatterns of the Hybrid Array with Two Sources at Different Frequencies and a Spatial Separation of 60° in θ Processed with Separate Center Frequencies

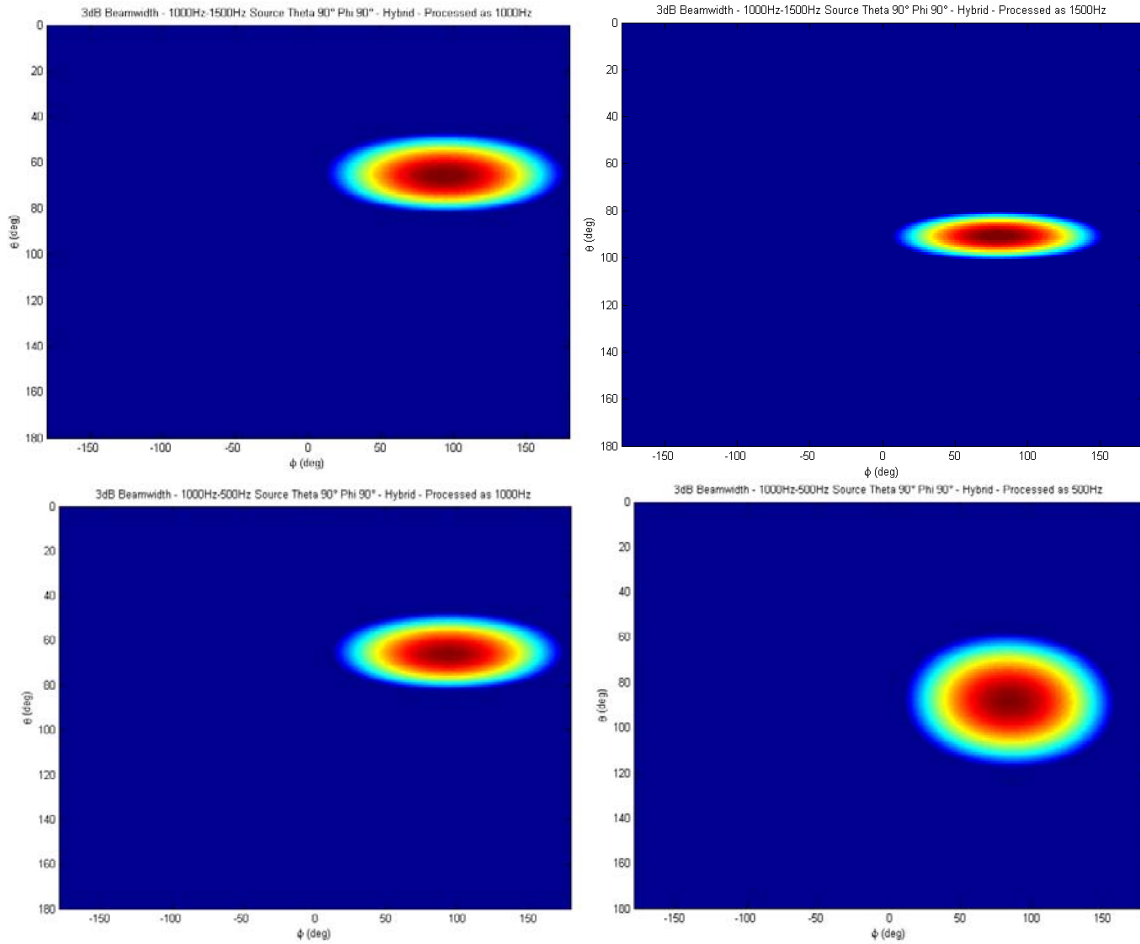


Figure 77. Beampatterns of the Hybrid Array with Two Sources at Different Frequencies and a Spatial Separation of 20° in ϕ Processed with Separate Center Frequencies

I. LIVE TARGET

A live trial was conducted as an attempt to track the aircraft that pass over the Naval Postgraduate School on final approach to Monterey Airport. Unfortunately, the day prior to the trial the array was damaged and one velocity element on one of the of the Microflown sensors was lost. The sensor and array were repaired as best as they could be. New alignment and transfer functions were performed and calculated. The live data was recorded with only eight

channels instead of nine. The lost element was one of the blue elements, which fortunately lies along the long axis of the array and least affected the array beamforming in this application.

Data was taken of several live targets on top of Spanagel Hall. Three jet aircraft and two turboprop aircraft were recorded. The signals were Fourier transformed and plotted to find the aircraft source tonals. Two of the five files contained high levels of low frequency broadband noise, most likely due to airflow from the light breeze blowing across the roof. No tonals were apparent on these files. Of the three remaining files, two corresponded to jet aircraft and one corresponded to a turboprop aircraft.

There was a significant signal to noise problem with the velocity elements. No discernable signal was present. The files were converted to audio and evaluated aurally. The pressure element signals clearly recorded the plane passing overhead; the velocity elements presented only wind noise. The data was recorded without the screen covers, as the array had been tested. Since the velocity elements operate by the passage of air across the sensing element, it is likely the wind blowing across the elements obscured any target signals. Future live tests should include windscreens covering the sensors.

The tonal frequencies extended from the extreme low frequency range into the midrange design frequency of the array. The tonal frequency content can be seen on the Aco microphone FFT plots, Figures 78 and 79. Figure 80 shows the presence of the target on the Aco pressure microphone and low signal to noise ratio obscuring the target on the velocity elements

Complicating the effort was the lack of coherence of the transfer functions in the low frequency region, Figures 81 and 82. Since the signal used to calculate the transfer functions was not accurately reproduced in the low frequency region, the coherence was very low. The response of various trials in the low frequency region was also erratic. Thus, the transfer functions cannot be relied on in this region. Obtaining new transfer functions using a signal

reproduced by a low frequency driver, would likely improve the low frequency performance of the beamformer. The output from the beamformer when processing the live target signals could not be discerned from the random noise of the same data files. Therefore, no conclusive results were produced. It is likely the array could track the aircraft with windscreen in place and using low frequency calibrated transfer functions.

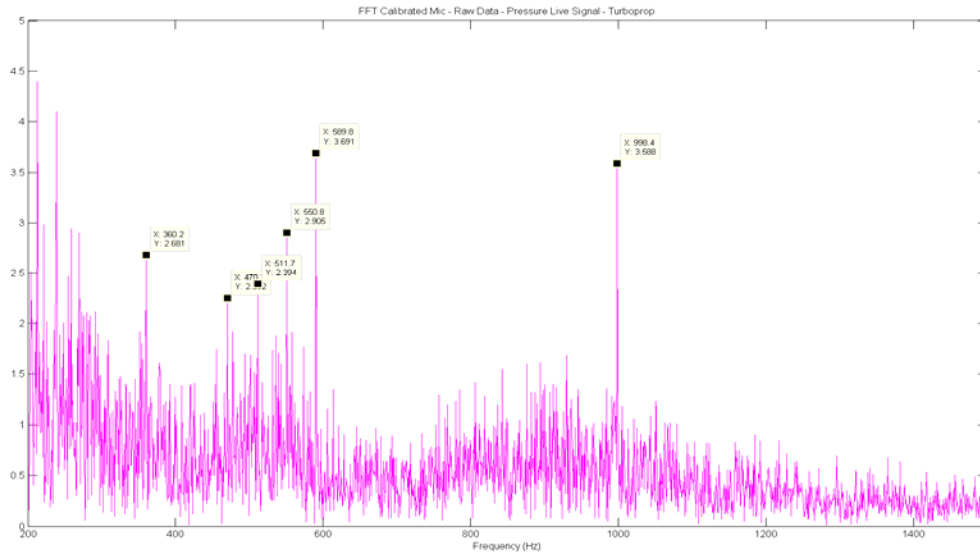


Figure 78. Frequency Content of the Live Signal – Turboprop Target

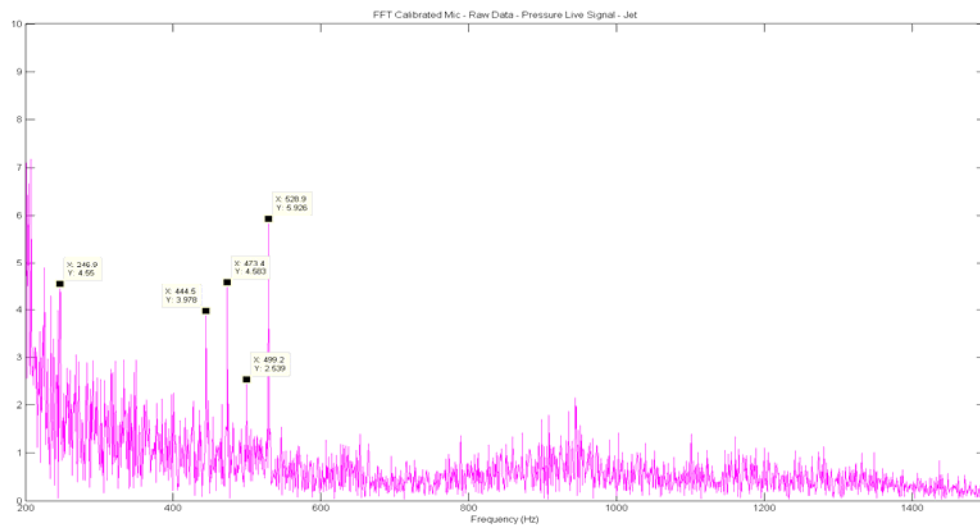


Figure 79. Frequency Content of the Live Signal – Jet Target

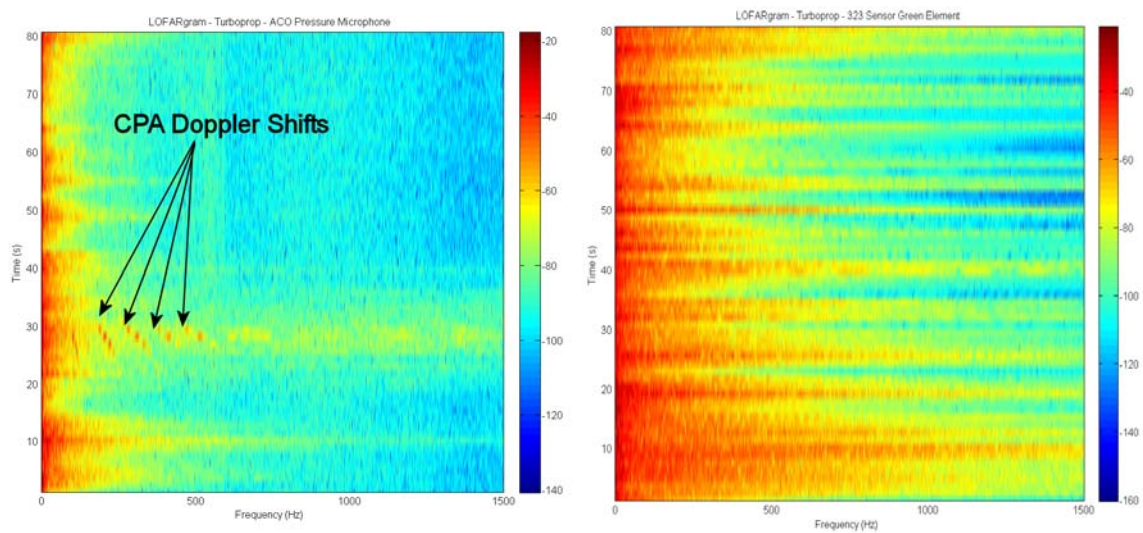


Figure 80. LOFARGrams of Turboprop Target for ACO Microphone and Sensor 323 Green Velocity Element

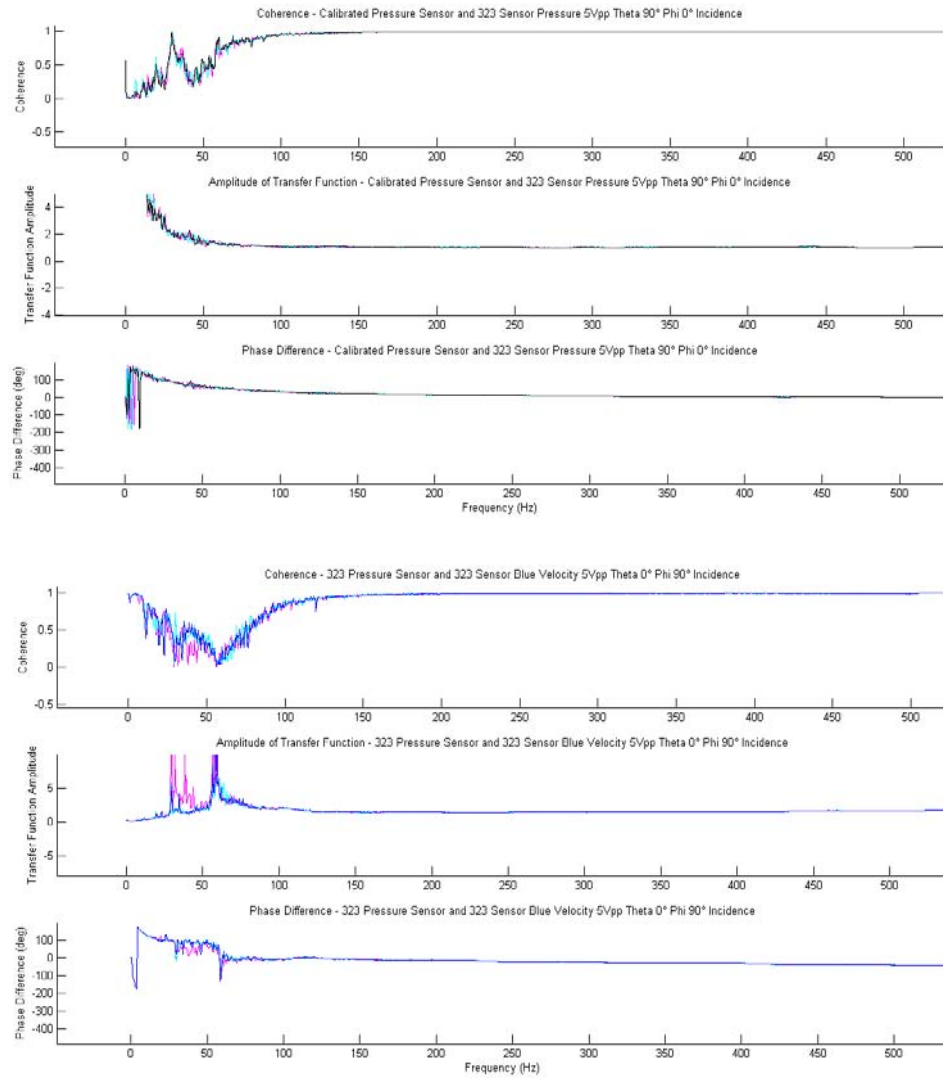


Figure 81. Low Frequency Region of Sample Pressure and Velocity Element Transfer Functions

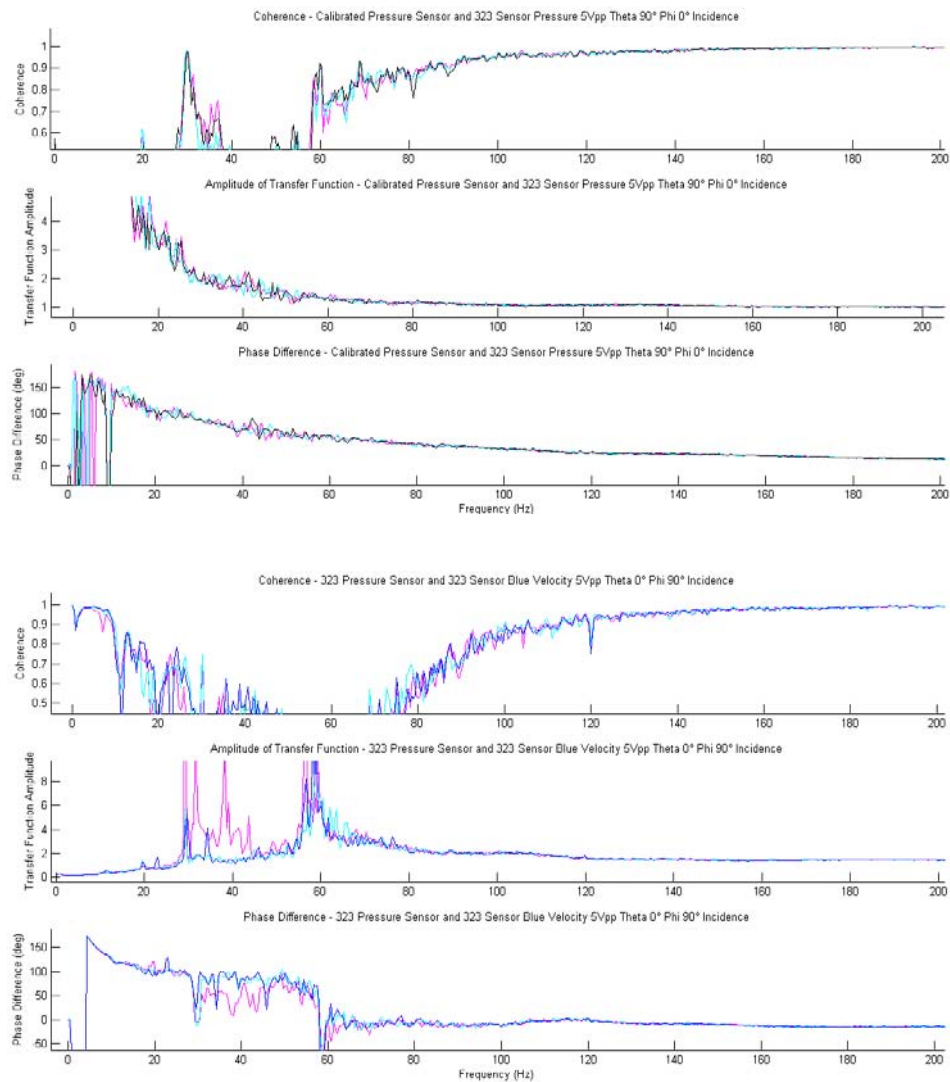


Figure 82. Erratic Response of Successive Trials in Low Frequency Region of Sample Pressure and Velocity Element Transfer Functions

VIII. CONCLUSION

The objectives outlined in the beginning of this project were met. The sensors were characterized and calibrated relative to a common reference using transfer functions. The Microflown sensors displayed no significant change in response when handling signals of different amplitude or aspect. The lack of absolute orthogonality of the three velocity elements did not produce noticeable errors in the beamformer. It did not detract from their ability to resolve direction to a target in three dimensions to within experimental accuracy. The simple linear summing algorithm with directional component weighting appears to work satisfactorily in the beamformer.

The directional nature of the Microflown vector sensors provided significant enhancements over a standard omnidirectional linear acoustic array. The instantaneous bearing resolution in three dimensions was the most significant.

The method of calibration here is also noteworthy. At no time were the sensors calibrated in a traditional sense by comparing the voltage outputs to a known pressure reference signal. The use of transfer functions to relate the outputs of the various elements to a single reference sensor in the frequency domain made the beamforming process straightforward and eliminated the need to explicitly calibrate the vector sensors.

It is possible that by simply adding additional elements and lengthening the array, there will be a corresponding improvement in bearing resolution due to wider angular separation to the ends of the array. Even better results are likely from construction of a planar or volumetric array using more sensors to gain an improvement on the inherent beamwidth of the vector sensors by taking advantage of increased spatial diversity. To achieve spatial separation in both angular planes, it may be possible to modify the three-element array by simply realigning the Microflown sensors so their MRAs are at 45° angles to the

longitudinal axis of the fixture. The only other modification would be new transfer functions in the changed alignment. The z and y axes of the coordinate system would rotate 45° with the realignment. This would create spatial separation between the sensors in both angular planes, with the ACO remaining at the origin.

Future improvements should include determining transfer functions in the low frequency region, repeating live tests with windscreens, and possibly using a more advanced beamforming algorithm to improve bearing resolution.

APPENDIX A. TECHNICAL SPECIFICATIONS

A. MICROFLOWN ULTIMATE SOUND PROBE



Charting sound fields

V1.0 2006-11-16

Datasheet Ultimate Sound Probe



Sensor configuration

- 3" Microflown Titan sensor element
- 1" miniature 1/10" pressure microphone

Physical characteristics

- diameter 1/2"
- length: 130 mm
- weight : 43gr
- removable slotted protective cap

Acoustical properties microphone element

- frequency range: 20Hz – 20kHz
- upper sound level 110dB (SPL re. 20µPa)
- sensitivity: 20mV/Pa
- directivity: omni
- signal to noise ratio (at 1kHz in 1Hz bandwidth): 105dB

Acoustical properties Microflown element

- frequency range: 1Hz – 20kHz
- upper sound level 135dB (PVL re. 50nm/s)
- sensitivity: 100mV/Pa²=20mV/(mm/s)
- directivity: figure of eight
- signal to noise ratio (at 1kHz in 1Hz bandwidth): 96dB

Electrical properties

- powering: 4 channel signal conditioner
- connector: LEMO 7 pins male

Environments

- temperature maximum 60 degrees Celsius

Microflown Technologies, PO Box 300, 6900 AH Zevenaar, The Netherlands
W: www.microflown.com E: info@microflown.com T: +31 316 581490 F: +31 316 581491

(From: Microflown Holland, n.d.)

Model Microflown

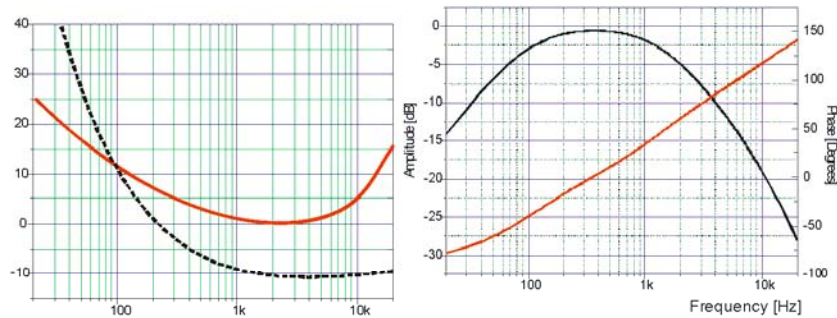
$$Sensitivity(f) = \frac{0.85}{\sqrt{1 + \left(\frac{112}{f}\right)^2} \sqrt{1 + \left(\frac{f}{1400}\right)^2} \sqrt{1 + \left(\frac{f}{14000}\right)^2}} \left[\frac{mV}{mm/s} \right]$$

$$phase = \tan^{-1} \frac{112}{f} - \tan^{-1} \frac{f}{1400} - \tan^{-1} \frac{f}{14000}$$

Model microphone

$$Sensitivity(f) = \frac{15}{\sqrt{1 + \left(\frac{35}{f}\right)^2}} \left[\frac{mV}{Pa} \right]$$

$$phase = \tan^{-1} \frac{35}{f}$$

Selfnoise and Intrinsic frequency response

(From: Microflown Holland, n.d.)

B. MICROFLOWN SIGNAL CONDITIONER



Charting sound fields

V1.2 2007-07-4

Datasheet

4 channel signal conditioner



Use

The 4 channel signal conditioner is used for the USP.

Channel nr. 1 is used for the pressure signal, channel 2 represents the blue element, nr. 3 the green and nr. 4 the red microflown element. The power switch can put the preamp in a LowGain state by pushing the switch down. When you put the switch up, the conditioner is in its HighGain state.

With the eq. switch the velocity signal can be equalized so that the frequency response becomes flat. When the led is orange the conditioner is in this "correction" mode. Push the switch up, the led will become green and the signal uncorrected.

The conditioner is powered with 18VDC.

Specifications

Weight:	0.75 kg.
Length:	120 mm
Width:	180 mm
Height:	45 mm
Connector output:	BNC
Connector input:	7 pin Lemo

Microflown Technologies, PO Box 300, 6900 AH Zevenaar, The Netherlands
W: www.microflown.com E: info@microflown.com T: +31 316 581490 F: +31 316 581491

(From: Microflown Holland, n.d.)

Technical specs. of signal conditioner



Channel nr. 1 is used for the pressure signal, channel 2 represents the blue element, nr. 3 (optional) the green and nr. 4 (optional) the red Microflown element.

The power switch can put the preamp in a LowGain state by pushing the switch down. When you put the switch up, the conditioner is in its HighGain state.

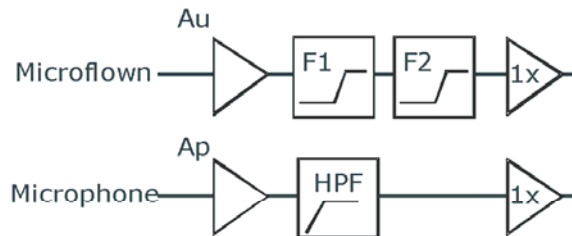
With the eq. switch the velocity signal can be equalized so that the frequency response becomes flat. When the led is orange the conditioner is in this "correction" mode. Push the switch up, the led will become green and the signal uncorrected.

The conditioner is powered standard with 18VDC.

(From: Microflown Holland, n.d.)

Technical specifications of the signal conditioner

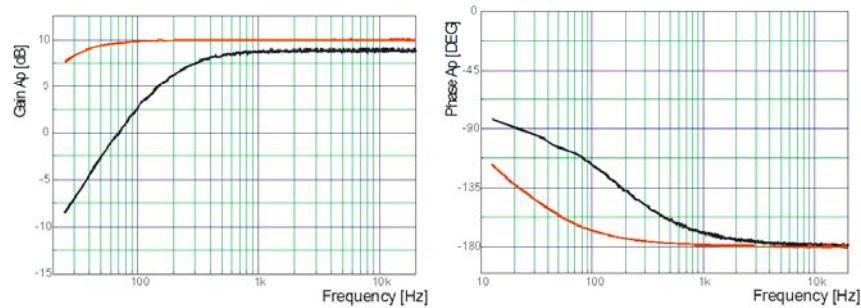
The signal conditioner consists of two types of preamplifiers: the microphone preamplifier and the Microflown preamplifier. A two channel signal conditioner consists of one microphone and one Microflown preamplifier, a four channel signal conditioner consists of one microphone and three Microflown preamplifiers.



The microphone preamplifier

The microphone preamplifier has an adjustable high pass filter (only to be set by a skilled engineer by opening the housing). The filter is set so that it matches the low frequency phase behavior of the Microflown sensor. (See also the document 'A standard calibration with a sphere calibrator'). The gain and the filter settings cannot be changed by the user.

Below the amplitude and phase response of the maximal settings are shown. Any value between these settings can be set.



(From: Microflown Holland, n.d.)

The Microflown preamplifier

The Microflown preamplifier has a dual adjustable equalizing filter (only to be set by a skilled engineer by opening the housing). The equalizing filter is set so that the Microflown response is corrected for higher frequencies ($f > 200\text{Hz}$). (See also the document 'A standard calibration with a sphere calibrator'). The gain of the preamplifier changes if the equalizer settings are changed.

The Microflown preamplifier has two gain settings that can be operated by the user. The low gain is used for high input signals of the Microflown and the high gain is set for low input signals. If the Microflown preamplifier is set in high gain it has good selfnoise properties: the noise of the preamp will not contribute to the selfnoise of the Microflown.

Allow a 10 seconds settling time after switching the Microflown preamplifier in high gain.

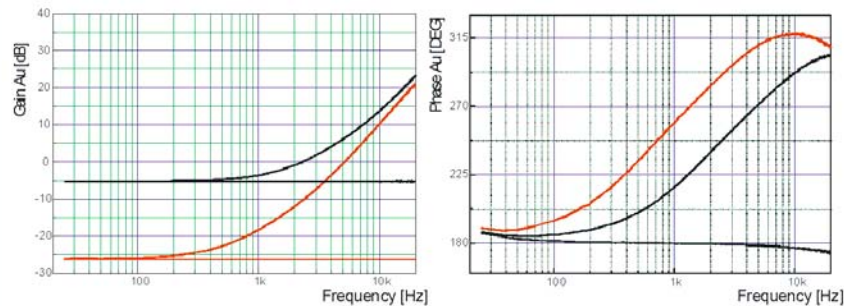
The equalizer can be switched on and off by the user.

The Microflown preamplifier can be set in high and low gain by the user

The graphs that show the behavior of the Microflown preamplifier are explained now.

The gain A_u [in dB] of the Microflown preamplifier varies with the setting of the equalizer. The graph below left shows this. The black line shows the minimal equalizer setting; this setting has a gain of approx -5dB (this is attenuation). If the equalizer is switched on (by the user), the gain increases from 1kHz. If the equalized is set maximal (red line), the gain is approx. -26dB (this is attenuation). If the equalizer is switched on (by the user), the gain increases from 200Hz.

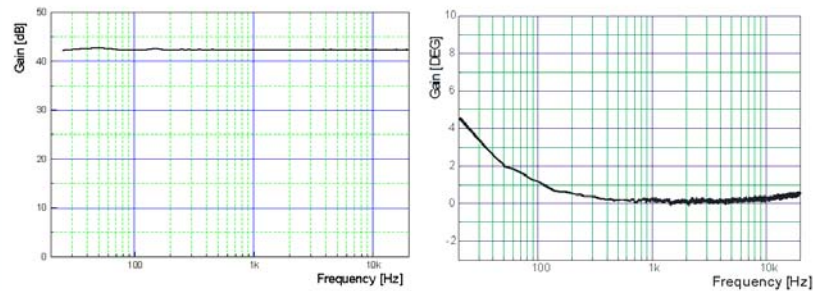
The phase response is approximately flat if the equalizer is not switched on, see figure right below, black lower line. If the equalizer is switched on, the phase response changes to match the opposite of the Microflown phase response.



Microflown Technologies, PO Box 300, 6900 AH Zevenaar, The Netherlands
 W: www.microflown.com E: info@microflown.com T: +31 316 581490 F: +31 316 581491

(From: Microflown Holland, n.d.)

The extra gain of the Microflown preamplifier can be switched on and off by the user. The gain is approx. 42dB independent on frequency. The phase response (right) is almost flat. At lower frequencies it rises a few degrees.



NB

The signal conditioner is normally used in High Gain

Input specifications

The signal conditioner is powered with 12VDC - 35VDC

The current at (nominal) 18VDC:

4 channel signal conditioner (/with USP):

High gain corrected: 32mA (/60mA)

High gain uncorrected: 28mA (/56mA)

Low gain corrected: 31mA (/59mA)

Low gain uncorrected: 27mA (/55mA)

2 channel signal conditioner (/with PU/Scanning):

High gain corrected: 18mA (/30mA / 29mA)

High gain uncorrected: 15mA (/27mA / 26mA)

Low gain corrected: 17.7mA (/29.7mA / 28.7mA)

Low gain uncorrected: 14.7mA (/26.7mA / 25.7mA)

Internal powering module: 3mA

Preamp (no correction circuit) with powering (3mA) and PU probe: 24mA

Maximal input and output voltage:

Microphone: maximal input: 0.9V_{tt}

Microflown: Maximal input low gain 350mV_{tt}

Maximal input high gain 10mV_{tt}

(From: Microflown Holland, n.d.)

C. ACO PACIFIC MICROPHONE

1. Cartridge Microphone Model 7046

CONDENSER MICROPHONE
FREQUENCY RESPONSE

CARTRIDGE TYPE 7046
SERIAL No. 36930

ACO

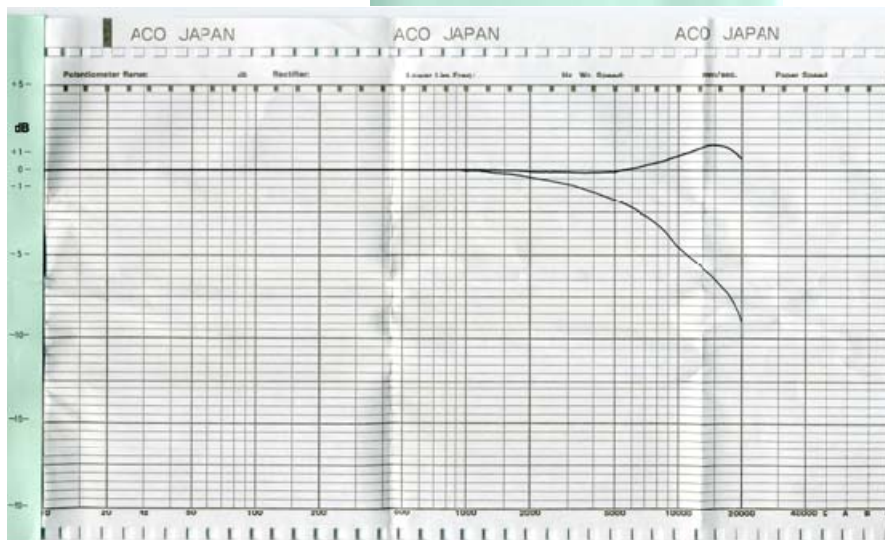
-25.3 dB re. 1 V per pascal or 54.33 mV per pascal

Open Circuit Correction Factor:
 $K_0 =$ dB

Capacitance:
 $C = 20.5$ pF

Leakage Resistance tested at 52% relative humidity
>10¹² Ω

Conditions of Test:
Frequency: 250 Hz
Polarization voltage: 200 V
Ambient Static Pressure: 1023 hPa
Relative Humidity: 40 %
Temperature: 20.0 °C
Date: MAR 24 2008
Signature: _____
1 pascal = 1 N/m² = 10 mbar



2. Preamplifier 1/2-inch Model 4012

4012 Family

Specifications: Typical

- 28 Vdc, 22 pF (DM2-22 Dummy Mic) Terminated
- Frequency Response:
<2 Hz to >200 kHz +/- 0.5 dB,
0.5 Hz to >> 200 kHz +/- 2 dB
- Insertion Loss: <<0.25 dB @ 1 kHz
- Noise:
"A" Weighted - <1.2 uV typ.
Linear 20 to 20 kHz - 3 uV
- Input Impedance: >50 G//0.1 pF
- Output Voltage: >7 Vrms @28 Vdc Typical
>15 Vrms @50 Vdc
- Power Requirements:
28 Vdc @ <1 mA (HP option <2 mA)
50 Vdc @ <2 mA (HP option <4 mA)



Models :

- 4012 w/CA4012 Cable (5 pin XLR)
- 4012L7 w/CA4012L7 Cable (7 pin Lemo)
- 40L12 - Lemo 1B 7 Pin Male Connector on base
- 40X12 - XLR 5 pin Male Connector on Base
- "HP" Version - Higher Operating Current Option
- Standard Cable Length 2 Meters
- 4016 - 4012 Preamp and AD0016 1/4" to 1/2" Adaptor
- 4022 - 4012 Preamp and AD0122 1" to 1/2" Adaptor

(From: ACO Pacific, n.d.)

D. NATIONAL INSTRUMENTS CHASSIS

NI CompactDAQ USB Data Acquisition System

NI CompactDAQ

NI cDAQ-9172

- Accepts up to 8 C Series I/O modules
- Compact (25 by 9 by 9 cm)
- Hi-Speed USB connection to PC
- 11 to 30 V power supply included

NI C Series Modules

- Sensor-specific signal conditioning per module
- Up to 24-bit resolution
- Up to 3.2 MS/s per chassis
- Up to 256 channels per chassis

Operating Systems

- Windows 2000/XP

Recommended Software

- LabVIEW
- LabWindows/CVI
- Measurement Studio
- VI Logger

Measurement Services Software (included)

- NI-DAQmx driver
- Measurement & Automation Explorer



Overview

National Instruments NI CompactDAQ brings the simplicity of USB to sensor and electrical measurements on the benchtop, in the field, and on the production line. By combining the ease of use and low cost of a data logger with the performance and flexibility of modular instrumentation, NI CompactDAQ delivers fast, accurate measurements in a small, simple, and affordable system. Flexible software options make it easy to use NI CompactDAQ to log data for simple experiments or to develop a fully automated test or control system. The modular design can measure up to 256 channels of electrical, physical, mechanical, or acoustical signals in a single system. In addition, per-channel analog-to-digital converters and individually isolated modules ensure fast, accurate, and safe measurements.

Key Features

- Small, modular data acquisition system for industrial measurements
- Hi-Speed USB plug-and-play connectivity to PC
- Hot-swappable, autodetectable C Series I/O modules with direct sensor and signal connectivity
- NI-DAQmx API that provides powerful and easy-to-use programming interface
- Up to 2,300 V_{rms} isolation (withstand)
- NI CompactDAQ environmental certifications and ratings:
 - 0 to 55 °C operating temperature
 - 30 g shock rating (operating)
- PXI vibration specification
- International safety, EMC, and environmental certifications

C Series Modules

Type of Signal	Signal	Module	Channels	Special Features	Connectivity
Analog Input	Thermocouple	NI 9211	4 DI	24-bit delta-sigma, 14 S/s, differential (J, K, R, S, T, N, E, and B thermocouple types)	Screw terminal
	IEPE sensors (accelerometer, microphone)	NI 9233	4 SE	24-bit, 50 kS/s, simultaneous, IEPE conditioning	BNC
	General-purpose (±200 mV to ±10 V)	NI 9205	32 SE/16 DI	16-bit, 250 kS/s	Spring terminal or D-Sub
		NI 9206	16 DI	16-bit, 250 kS/s, 600 VDC Cat. I bank isolation	Spring terminal
		NI 9215	4 DI	16-bit, 100 kS/s/ch, simultaneous, differential	Screw terminal or BNC
	General-purpose (±80 mV)	NI 9211	4 DI	24-bit, 14 S/s, differential	Screw terminal
	Bridge	NI 9237	4	24-bit, 50 kS/s/ch	RJ 45
Analog Output	General-purpose (±10 V)	NI 9263	4 SE	16-bit, 100 kS/s/ch, simultaneous	Screw terminal
Digital Input	Bidirectional 5 V TTL	NI 9401	8	5 V TTL, ultrahigh-speed, bidirectional, 30 V protection	25-pin D-Sub
	24 V sinking	NI 9421	8	10 kS/s, 24 V logic, 40 V protection	Screw terminal or 25-pin D-Sub
Digital Output	Bidirectional 5 V TTL	NI 9401	8	5 V TTL, ultrahigh-speed, bidirectional, 30 V protection	25-pin D-Sub
	24 V sourcing	NI 9472	8	10 kS/s, 24 V logic, 750 mA max per ch, 30 V protection, short-circuit-proof	Screw terminal or 25-pin D-Sub
Relay	Electromechanical, Form A (SPST)	NI 9481	4	30 VDC (2 A), 60 VDC (1 A), 250 VAC (2 A) electromechanical Form A (SPST)	Screw terminal
Counter, Pulse Generation	Counter/timer/PWM/pulse generation (TTL)	NI 9401	8	5 V TTL, ultrahigh-speed, bidirectional, 30 V protection	25-pin D-Sub
	PWM/pulse generation (24 V)	NI 9472	8	10 kS/s, 24 V logic, 750 mA max per ch, 30 V protection, short-circuit-proof	Screw terminal or 25-pin D-Sub



(From: National Instruments, n.d.)

NI CompactDAQ USB Data Acquisition System

C Series modules accept a broad range of I/O including thermocouple inputs, ±10 V simultaneous-sampling analog input, ±10 V analog output, IEPE sensors, bridge inputs, 24 V industrial digital I/O with up to 750 mA current drive, and 5 V/TTL digital I/O. Each C Series module contains built-in signal conditioning and screw terminal, BNC, or D-Sub connectors so you can connect signals directly. This direct connectivity, in turn, significantly reduces space requirements and field-wiring costs.

Driver Software

NI-DAQmx is the driver software included with NI CompactDAQ and most National Instruments data acquisition and signal conditioning products. This easy-to-use software integrates the full functionality of your NI CompactDAQ hardware with National Instruments LabVIEW, LabWindows/CVI, or Measurement Studio for Visual Basic. High-performance features include multidevice synchronization, networked measurements, and simulated devices. Bundled with NI-DAQmx, the Measurement & Automation Explorer utility simplifies the configuration of your measurement hardware with device test panels, interactive measurements, and scaled I/O channels. NI-DAQmx also provides numerous example programs for LabVIEW and other application development environments to get you started with your application quickly.

Services and Support/Training

As a complement to your NI CompactDAQ system, consider:

- **Technical Support** – included in hardware/software purchase through applications engineers worldwide, and features Web resources with more than 3,000 example programs and more than 7,000 knowledge bases, and Premier Support – ni.com/support
- **Calibration** – NIST-traceable basic calibration certificate available, services for ANSI/NCSL-Z540 and periodic calibration – ni.com/calibration
- **Extended Warranty** – meet project life-cycle requirements and maintain optimal performance in a cost-effective way – ni.com/services
- **Data Acquisition Training** – take advantage of instructor-led courses – ni.com/training
- **Professional Services** – obtain feasibility assessments, consulting, and integration through National Instruments Alliance Partner program members – ni.com/alliance

For more information on NI services and support, visit ni.com/services.

Chassis	Module Slots	Connection to PC	Channels per Chassis	Analog Resolution	Analog Sampling Rate	Analog Throughput
cDAQ-9172	8	Hi-Speed USB	Up to 256 analog input, or 32 analog output, or 64 digital I/O	Up to 24 bits	Up to 400 kS/s per module	3.2 MS/s total throughput

Table 1. cDAQ 9172 Channel, Speed, and Resolution Specifications

Ordering Information

NI cDAQ-9172	
U.S., 120 VAC	779508-01
Switzerland, 220 VAC	779508-02
Australia, 240 VAC	779508-03
Europe, 240 VAC	779508-04
UK, 240 VAC	779508-06
Japan, 100 VAC	779508-07

BUY NOW!

For complete product specifications, pricing, and accessory information, call (800) 813 3693 or go to ni.com/compactdaq.

Specifications

Power

Input 11 to 30 VDC at 15 W

Physical Dimensions

cDAQ-9172 chassis	254 by 88.1 by 88.1 mm (10.0 by 3.47 by 3.47 in.)
External AC adapter	117 by 38.1 by 38.1 mm (4.60 by 1.50 by 1.50 in.)
C Series I/O modules	69.9 by 22.2 by 87.4 cm (2.75 by .875 by 3.44 in.)

Connectors

DC power input	2 conductor input jack with threaded retainer
C Series module I/O	Removable screw/spring terminal, BNC, or D-Sub connector
C Series module to chassis	15-pin female D-Sub connector

BUY ONLINE at ni.com or CALL (800) 813 3693 (U.S.)

(From: National Instruments, n.d.)

E. NATIONAL INSTRUMENTS DATA ACQUISITION MODULES

Sound and Vibration Data Acquisition

NI 9233, NI 9234 **NEW!**

- 24-bit resolution
- 102 dB dynamic range
- 4 simultaneous analog inputs
- ± 5 V input range
- Antialiasing filters
- TEDS read/write
- Supported in NI CompactDAQ, CompactRIO, and Hi-Speed USB carrier

Recommended Software

- LabVIEW
- LabVIEW Sound and Vibration Toolkit
- Sound and Vibration Measurement Suite



Model	Max Sampling Rate	IEPE	Coupling
NI 9233	50 kS/s	Always enabled (2 mA)	AC coupling
NI 9234	51.2 kS/s	Software selectable (0 or 2 mA)	Software selectable (AC or DC coupling)

Table 1. C Series Dynamic Signal Acquisition Selection Guide

Overview

The National Instruments 9233 and 9234 are four-channel dynamic signal acquisition modules for making high-accuracy measurements from IEPE sensors. The NI 9233 and 9234 C Series analog input modules deliver 102 dB of dynamic range and incorporate IEPE (2 mA constant current) signal conditioning for accelerometers and microphones. The four input channels simultaneously acquire at rates from 2 to 50 kHz or, with the NI 9234, up to 51.2 kS/s. In addition, the modules include built-in antialiasing filters that automatically adjust to your sampling rate. Compatible with a single-module USB carrier and NI CompactDAQ and CompactRIO hardware, the NI 9233 and 9234 are ideal for a wide variety of mobile/portable applications such as industrial machine condition monitoring and in-vehicle noise, vibration, and harshness testing.

Hardware

Analysis Capabilities	
Power spectra	Each simultaneous signal is buffered, analog prefiltered, and sampled by a 24-bit delta-sigma analog-to-digital converter (ADC) that performs digital filtering with a cutoff frequency that automatically adjusts to your data rate. The NI 9233 and 9234 feature a voltage range of ± 5 V and a dynamic range of more than 100 dB. In addition, the modules include the capability to read and write to transducer electronic data sheet (TEDS) Class 1 smart sensors. The NI 9233 and 9234 provide ± 30 V of
Zoom FFTs	
Fractional octave analysis	
Vibration level measurements	
Order spectra	
Transient analysis	

overvoltage protection (with respect to chassis ground) for IEPE sensor connections. The NI 9234 has three software-selectable modes of measurement operation: IEPE-on with AC coupling, IEPE-off with AC coupling, and IEPE-off with DC coupling. IEPE excitation and AC coupling are not software-selectable and are always enabled for the NI 9233.

The NI 9233 and 9234 use a method of A/D conversion known as delta-sigma modulation. If, for example, the data rate is 25 kS/s, then each ADC actually samples its input signal at 3.2 MS/s (128 times the data rate) and produces samples that are applied to a digital filter. This filter then expands the data to 24 bits, rejects signal components greater than 12.5 kHz (the Nyquist frequency), and digitally resamples the data at the chosen data rate of 25 kS/s. This combination of analog and digital filtering provides an accurate representation of desirable signals while rejecting out-of-band signals. The built-in antialiasing filters automatically adjust themselves to discriminate between signals based on the frequency range, or bandwidth, of the signal.



(From: National Instruments, n.d.)

Sound and Vibration Data Acquisition

USB Platform

The NI Hi-Speed USB carrier makes portable data acquisition easy. Simply plug the NI 9233 or 9234 into the USB carrier and begin acquiring data. Communication to the USB carrier is over Hi-Speed USB, guaranteeing data throughput.



NI CompactDAQ Platform

NI CompactDAQ delivers the simplicity of USB to sensor and electrical measurements on the benchtop, in the field, and on the production line. By combining the ease of use and low cost of a data logger with the performance and flexibility of modular instrumentation, NI CompactDAQ offers fast, accurate measurements in a small, simple, and affordable system. Flexible software options make it easy to use NI CompactDAQ to log data for simple experiments or to develop a fully automated test or control system. The modular design can measure up to 256 channels of electrical, physical, mechanical, or acoustical signals in a single system. In addition, per-channel ADCs and individually isolated modules ensure fast, accurate, and safe measurements.



NI CompactRIO Platform

When used with the small, rugged CompactRIO embedded control and data acquisition system, NI C Series analog input modules connect directly to reconfigurable I/O (RIO) field-programmable gate array (FPGA) hardware to create high-performance embedded systems. The reconfigurable FPGA hardware within CompactRIO provides a variety of options for custom timing, triggering, synchronization, filtering, signal processing, and high-speed decision making for all C Series analog input modules. For instance, with CompactRIO, you can implement custom triggering for any analog sensor type on a per-channel basis using the flexibility and performance of the FPGA and the numerous arithmetic and comparison function blocks built into NI LabVIEW FPGA.



BUY ONLINE at ni.com or CALL 800 813 3693 (U.S.)

(From: National Instruments, n.d.)

Analysis Software

The NI 9233 and 9234 are well-suited for noise and vibration analysis applications. The NI Sound and Vibration Measurement Suite, which specifically addresses these applications, has two components: the NI Sound and Vibration Assistant and LabVIEW analysis VIs (functions) for power spectra, frequency response (FRF), fractional octave analysis, sound-level measurements, order spectra, order maps, order extraction, sensor calibration, human vibration filters, and torsional vibration.

NI Sound and Vibration Assistant

The Sound and Vibration Assistant is interactive software designed to simplify the process of acquiring and analyzing noise and vibration signals by offering:

- A drag-and-drop, interactive analysis and acquisition environment
- Rapid measurement configuration
- Extended functionality through LabVIEW

Interactive Analysis Environment

The Sound and Vibration Assistant introduces an innovative approach to configuring your measurements using intuitive drag-and-drop steps. Combining the functionality of traditional noise and vibration analysis software with the flexibility to customize and automate routines, the Sound and Vibration Assistant can help you streamline your application.

Rapid Measurement Configuration

There are many built-in steps available for immediate use in the Sound and Vibration Assistant. You can instantly configure a measurement and analysis application with:

- Hardware I/O – generation and acquisition of signals from a variety of devices, including data acquisition devices and modular instruments
- Signal processing – filtering, windowing, and averaging
- Time-domain analysis – sound- and vibration-level measurements
- ANSI and IEC fractional-octave analysis
- Frequency-domain analysis – power spectrum, frequency response, power-in-band, peak search, and distortion
- Order analysis – tachometer processing, order power spectrum, order tracking, and order extraction
- Report generation – ability to drag and drop signals to Microsoft Excel or export data to Microsoft Word or UFF58 files

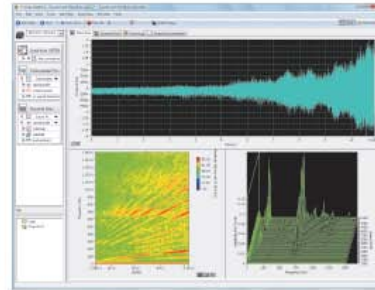


Figure 1. NI Sound and Vibration Assistant Performing Engine Run-up Test

Extended Functionality through LabVIEW

Reuse your measurement applications developed with the Sound and Vibration Assistant in LabVIEW by converting projects into LabVIEW block diagrams. With the LabVIEW full-featured graphical programming environment, you can further automate your application or customize your analysis.

Sound and Vibration Analysis VIs for LabVIEW

With the sound and vibration analysis VIs in LabVIEW, you can develop a variety of custom audio, acoustic, and vibration applications. Functionality includes:

- Full, 1/3, 1/6, 1/12, and 1/24 octave analysis with linear A, B, or C weighting
- Baseband, zoom, and subset power spectrum
- Peak search and Power in band
- Frequency response (FRF)
- Filtering
- Swept sine
- Distortion analysis (THD, THD+N, IMD)
- Noise measurements (SNR)
- Human vibration weighting filters
- Torsional vibration
- Tachometer signal processing
- Order tracking, spectrum, and Order extraction
- Waterfall display for power, octave, and order spectra
- Shaft centerline, orbit, Bode, and polar plot format
- File input and output to UFF58

BUY ONLINE at ni.com or CALL 800 813 3693 (U.S.)

(From: National Instruments, n.d.)

Sound and Vibration Data Acquisition

Recommended Hardware

The Sound and Vibration Measurement Suite includes more than 50 examples that work with both dynamic signal acquisition (DSA) and multifunction data acquisition devices. For sound and vibration data acquisition, National Instruments recommends DSA devices. With 24-bit ADCs and digital-to-analog converters (DACs) and integrated antialiasing filters, DSA devices are ideal for acoustic, noise, and vibration measurements.

There are numerous system requirements to consider when selecting data acquisition hardware for measuring or generating sound and vibration signals. From IEPE signal conditioning for accelerometers and microphones to high dynamic range (up to 118 dB) and multichannel synchronization (up to 13,000 channels), National Instruments offers a wide range of hardware products for your applications.

Product	Bus	Input Resolution (bits)	Dynamic Range (dB)	Sampling Rate per Channel	Analog Inputs	Input Range	Gain Settings	Coupling	TEDS Support	Analog Outputs
High Performance										
NI 4461	PXI, PCI	24	118	204.8 kS/s	2	±42 V to 316 mV	-20 to 30 dB in 10 dB increments	AC/DC	✓	2
NI 4462	PXI, PCI	24	118	204.8 kS/s	4	±42 V to 316 mV	-20 to 30 dB in 10 dB increments	AC/DC	✓	—
High Density										
NI 4496	PXI	24	114	204.8 kS/s	16	±10 to 1 V	0 to 20 dB	DC	—	—
NI 4496	PXI	24	114	204.8 kS/s	16	±10 to 1 V	0 to 20 dB	AC	✓	—
NI 4498	PXI	24	114	204.8 kS/s	16	±10 V to 316 mV	0 to 20 dB	AC	✓	—
Low Cost										
NI 4472	PXI, PCI	24	110	102.4 kS/s	8	±10 V	—	AC/DC	—	—
NI 4474	PCI	24	110	102.4 kS/s	4	±10 V	—	AC/DC	—	—
Ultraportable										
NI 9233	USB	24	102	50 kS/s	4	±5 V	—	AC	✓	—
NI 9234	USB	24	102	51.2 kS/s	4	±5 V	—	AC/DC	✓	—

Table 2. Additional NI Dynamic Signal Acquisition Devices

Ordering Information

NI 9233	779015-01
NI 9233 with Sound and Vibration Measurement Suite	779015-02
NI USB-9233 with USB carrier	779365-01
NI USB-9233 with Sound and Vibration Measurement Suite	779366-01
NI 9234	779680-01
NI 9234 with Sound and Vibration Measurement Suite	779680-02
NI USB-9234	780235-01
NI USB-9234 with Sound and Vibration Measurement Suite	780235-02

BUY NOW!

For complete product specifications, pricing, and accessory information, call 800 813 3693 (U.S.) or go to ni.com/soundandvibration.

BUY ONLINE at ni.com or **CALL** 800 813 3693 (U.S.)

(From: National Instruments, n.d.)

Sound and Vibration Data Acquisition

NI 9233 Specifications

>>For complete specifications, see the **NI 9233 Operating Instructions and Specifications** at ni.com/manuals.

The following specifications are typical for the range 0 to 60 °C unless otherwise noted.

Input Characteristics

Number of channels.....	4 analog input
ADC resolution	24 bits
Type of ADC	Delta-sigma (with analog prefiltering)

Data rate (fs)	
Minimum	2 kS/s
Maximum	50 kS/s

Master timebase (internal)	
Frequency.....	12.8 MHz
Accuracy	±100 ppm max
Input coupling	AC

AC cutoff frequency	
-3 dB.....	0.5 Hz typ
-0.1 dB.....	4.2 Hz max

AC voltage full-scale range	
Typical	5.4 V _{pk}
Minimum	5 V _{pk}
Maximum	5.8 V _{pk}

Common-mode voltage (AI- to earth ground).....	±2 V
---	------

IEPE excitation current	
Minimum	2.0 mA
Typical	2.2 mA
IEPE compliance voltage	19 V max

Overvoltage protection (with respect to chassis ground)	
For an IEPE sensor connected to AI+ and AI-	±30 V
For a low-impedance source connected to AI+ and AI-	-6 to 30 V

Accuracy (0 to 60 °C)

Error	Accuracy
Calibrated max	±0.3 dB
Calibrated typ	±0.1 dB
Uncalibrated max	±0.6 dB

NI 9234 Specifications

>>For complete specifications, see the **NI 9234 Specifications** at ni.com/manuals.

The following specifications are typical for the range 0 to 60 °C unless otherwise noted.

Input Characteristics

Number of channels.....	4 analog input
ADC resolution	24 bits
Type of ADC	Delta-sigma (with analog prefiltering)

Data rate (fs)	
Minimum	1.65 kS/s
Maximum	51.2 kS/s

Master timebase (internal)	
Frequency.....	13.1 MHz
Accuracy	±50 ppm max
Input coupling	Software-selectable AC/DC

AC cutoff frequency	
-3 dB.....	0.5 Hz typ
-0.1 dB.....	4.6 Hz max

AC voltage full-scale range	
Typical	5.1 V _{pk}
Minimum	5 V _{pk}
Maximum	5.2 V _{pk}

Common-mode voltage (AI- to earth ground).....	±2 V
---	------

IEPE excitation current	
Minimum	2.0 mA
Typical	2.1 mA
IEPE compliance voltage.....	19 V max

Overvoltage protection (with respect to chassis ground)	
For an IEPE sensor connected to AI+ and AI-	±30 V
For a low-impedance source connected to AI+ and AI-	-6 to 30 V

Accuracy (0 to 60 °C)

Error	Accuracy
Calibrated max	±0.3 dB
Calibrated typ	±0.002 dB
Uncalibrated max	±0.16 dB

BUY ONLINE at ni.com or CALL 800 813 3693 (U.S.)

(From: National Instruments, n.d.)

F. VIFA LOUDSPEAKER DRIVER



Vifa-Speak A/s
DK-6920 Videbæk
Denmark

Fabrikation af kvalitetshøjttalere
Production of Quality Loudspeakers

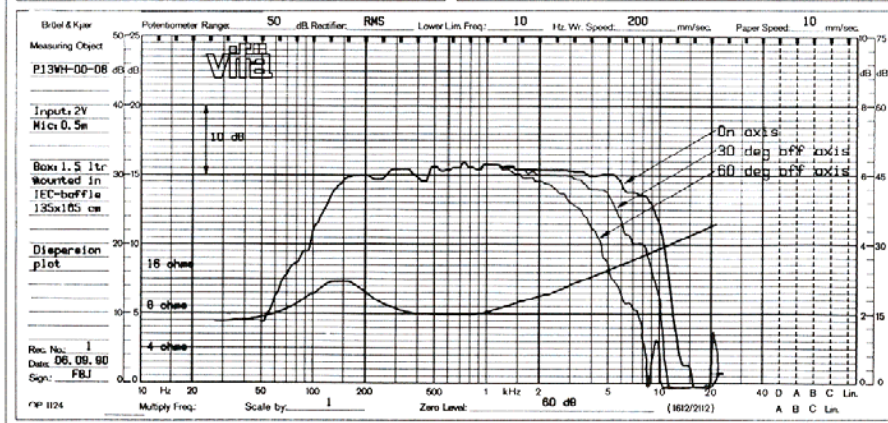
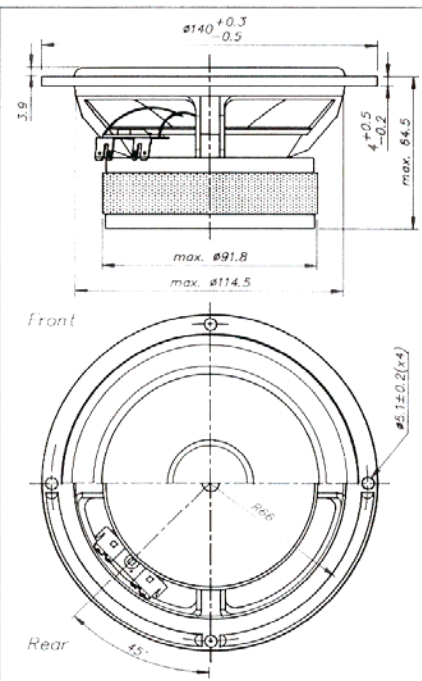
5" WOOFER

SPECIAL FEATURES:

- MINERAL FILLED POLYCONE
- HIGH DAMPING RUBBER SURROUND
- SMOOTH ROLL OFF
- OPTIMIZED OFF AXIS RESPONSE
- NEUTRAL MIDRANGE REPRODUCTION
- VENTED THROUGH V.C. FORMER AND POLE PIECE

NOMINAL IMPEDANCE	8 Ω
NOMINAL POWER (IEC 268-5)	40 W
MUSIC POWER (DIN 45500)	70 W
FREQUENCY RANGE	60-5000 Hz
SENSITIVITY (1W, 1m)	88 dB
EFFECTIVE CONE AREA	86 cm ²
VOICE COIL RESISTANCE	5,7 Ω
VOICE COIL INDUCTANCE	0,7 mH
OPERATING POWER	6,3 W
VOICE COIL DIAMETER	25 mm
VOICE COIL HEIGHT	14 mm
AIR GAP HEIGHT	6 mm
FREE AIR RESONANCE	60 Hz
MOVING MASS (incl. air)	7,5 g
FORCE FACTOR, B x l	6,0 Txm
MAGNET WEIGHT (14,6 oz)	415 g
Qms	1,38
Qes	0,43
Qts	0,33
Vas	10 ltr

P13WH-00-08



Stationsvej 5 • Postbox 39 • DK-6920 Videbæk • Tlf. 97 17 17 22 • Int. tel.: +45 97 17 17 22 • Telefax: +45 97 17 16 96
Postgiro 9 03 79 50 • Akts. reg.nr.: 209848 • Bank: A/S Ringkjøbing Bank, Videbæk afd. • SWIFT address RIBA DK 22, Telex: 62442

(From: Parts Express, n.d.)

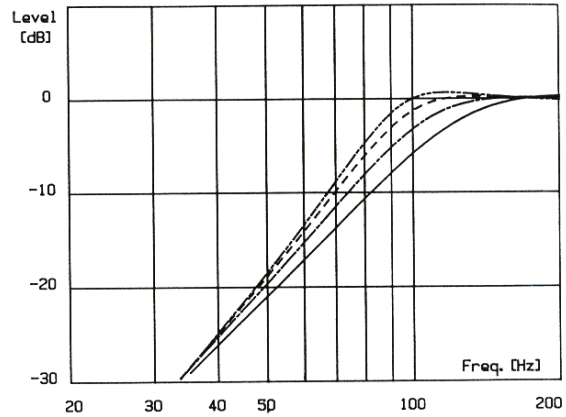
P13WH-00-08

Applications

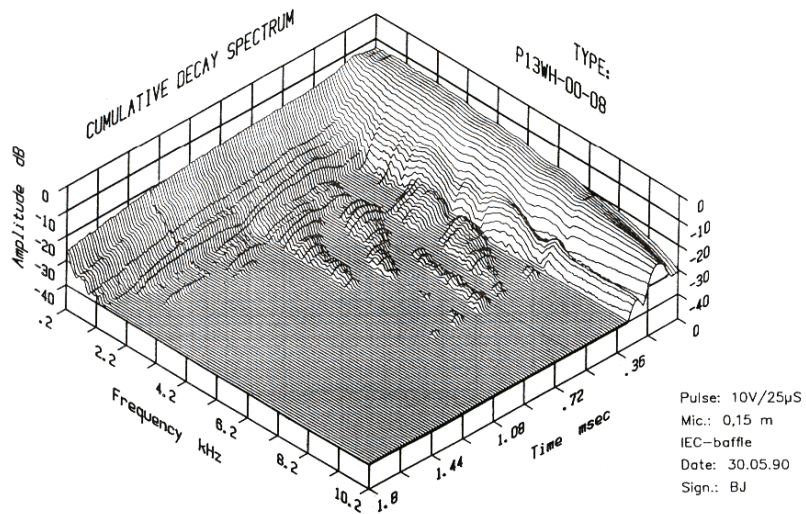
BASS REFLEX SYNTHESIS

	Vb [litre]	Fb [Hz]	Dp [cm]	Lp [cm]
1—	3	90	5.0	14.3
2---	5	90	6.0	11.4
3---	8	90	6.0	5.9
4---	12	90	7.0	4.5

Vb: Box Volume
Fb: Tuning Frequency
Dp: Port Diameter
Lp: Port Length



The P13WH-00-08 is designed as a small "general purpose" woofer. It may be used in 3-10 litres sealed enclosures, but the best result will be obtained in a 3-12 litres vented box. Another application is as midrange for very low crossover frequencies.



Stationsvej 5 • Postbox 39 • DK-6920 Videbæk • Tlf. 97 17 17 22 • Int. tel.: +45 97 17 17 22 • Telefax: +45 97 16 96
Postgiro 9 03 79 50 • Akts. reg.nr.: 209648 • Bank: A/S Ringkøbing Bank, Videbæk afd. • SWIFT address RIBA DK 22, Telex: 62442

(From: Parts Express, n.d.)

G. CALCULATED INDUSTRIES LASER DISTANCE METER



PREXISO X2
LASER DISTANCE MEASURER

Specifications
 Measuring range: 4" (10 cm) to 100' (30.5 m)
 Accuracy: $\pm 1/8"$ (3 mm) or better over entire range*
 Laser Class: II
 Units: Imperial/Metric
 Measuring Modes: Measures from front or rear edge in standard or continuous measure mode
 Weight: (product only with battery) 3.5 oz. (100 g)
 Power: 2 AAA batteries (included)
 Power Management: Auto shut-off after 3 minutes
 Battery life: Up to 3000 measurements
 Size: 4-3/4 x 2 x 1" (123 x 50 x 26 mm)
 Includes: Rugged carrying case, 2 AAA batteries and 3-year warranty
 Operating temperature range: 32°F to 104°F (0°C to 40°C)
 Country of Origin: Hungary

Calculated Industries
 4540 Hytech Drive, Carson City, NV 89706
 1-775-885-4900 • Fax: 1-775-885-4949 • www.calculated.com

True Laser Measuring — Easy to Use, Fast, Affordable and Precise

PREXISO X2
LASER DISTANCE MEASURER

The **PREXISO X2** is a one-person laser measuring tool for distances, areas and volumes. It is easier and faster to use than a tape measure, and more accurate than ultrasonic measures. Great for measuring in crowded rooms or measuring inaccessible areas. It is about the size of a modern cell phone, and fits easily in your hand, pocket or tool belt.

- Measure up to 100' with accuracy of 1/8" or better over entire range
- Faster and more accurate than tape measures
- Small size makes it easy to use in hard-to-access areas
- Measure distances in feet-inch-fractions, inch-fractions, or meters

For — Real Estate Agents and Brokers, Appraisers, Contractors/Builders, Carpenters, Plumbers, Interior Designers, Remodelers, Estimators, Do-It-Yourselfers, Painters — *Anyone Who Needs Laser Accurate Measurements*

Here are a few of the hundreds of uses for the Prexiso X2

- Measure distances, areas and volumes for material estimates
- Determine placement and fit for room furnishings and cabinetry
- Measure windows and window coverings
- Measure hallways, rooms, ceiling heights, and floors for floor covering
- Measure in attics and crawlspaces for running wire or insulation and ductwork
- Pythagoras functions — calculate distance, height or width; measure two sides of a triangle and it calculates the third

*In favorable conditions (optimal target surface, room temperature) for measurements up to 33 feet (10 m) accuracy is $\pm 1/16"$ (1.5 mm). In unfavorable conditions (intense sunlight, weak reflective surfaces or large temperature variations), accuracy may deteriorate to approximately $\pm 1/8"$ (3.2 mm) for distances above 33 ft. (10 m).

AREA



$L \times W = \text{Area}$

VOLUME



$L \times W \times H = \text{Volume}$

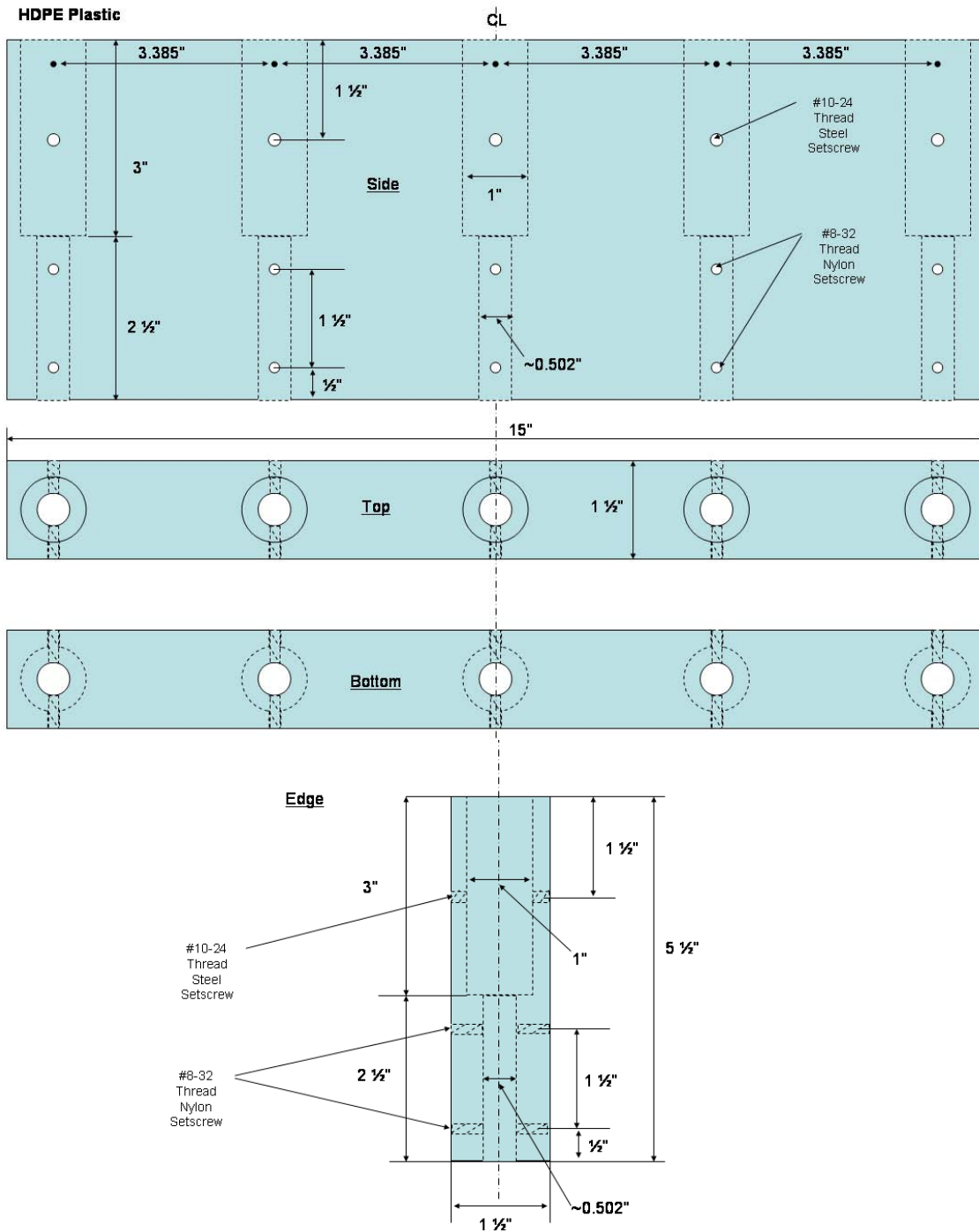
PYTHAGOREAN
(Indirect Measurements)

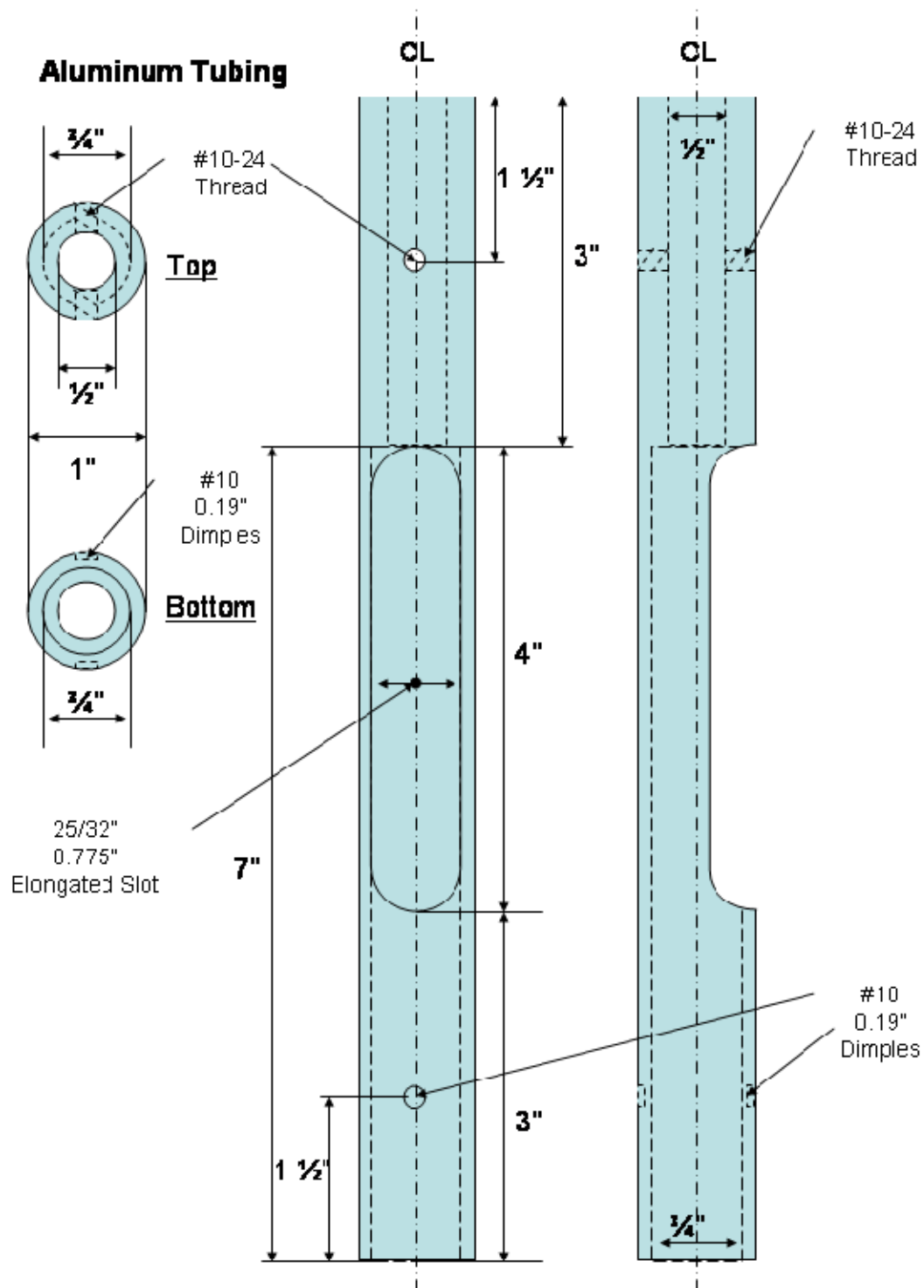


Measure to Top
Measure to Bottom
Solves for Height

(From: Calculated Industries, n.d.)

H. ARRAY SENSOR FIXTURE MACHINING PARAMETERS





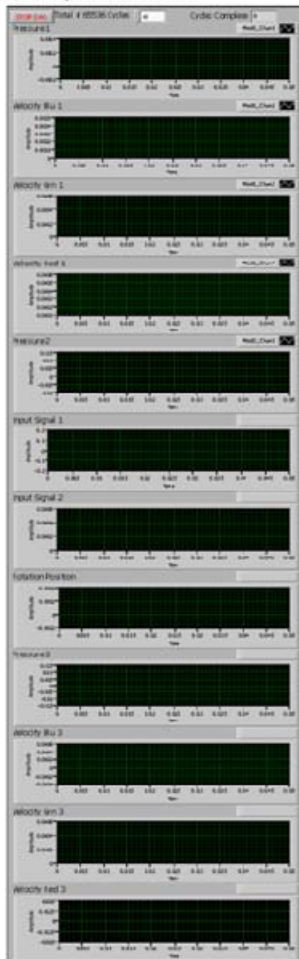
THIS PAGE INTENTIONALLY LEFT BLANK

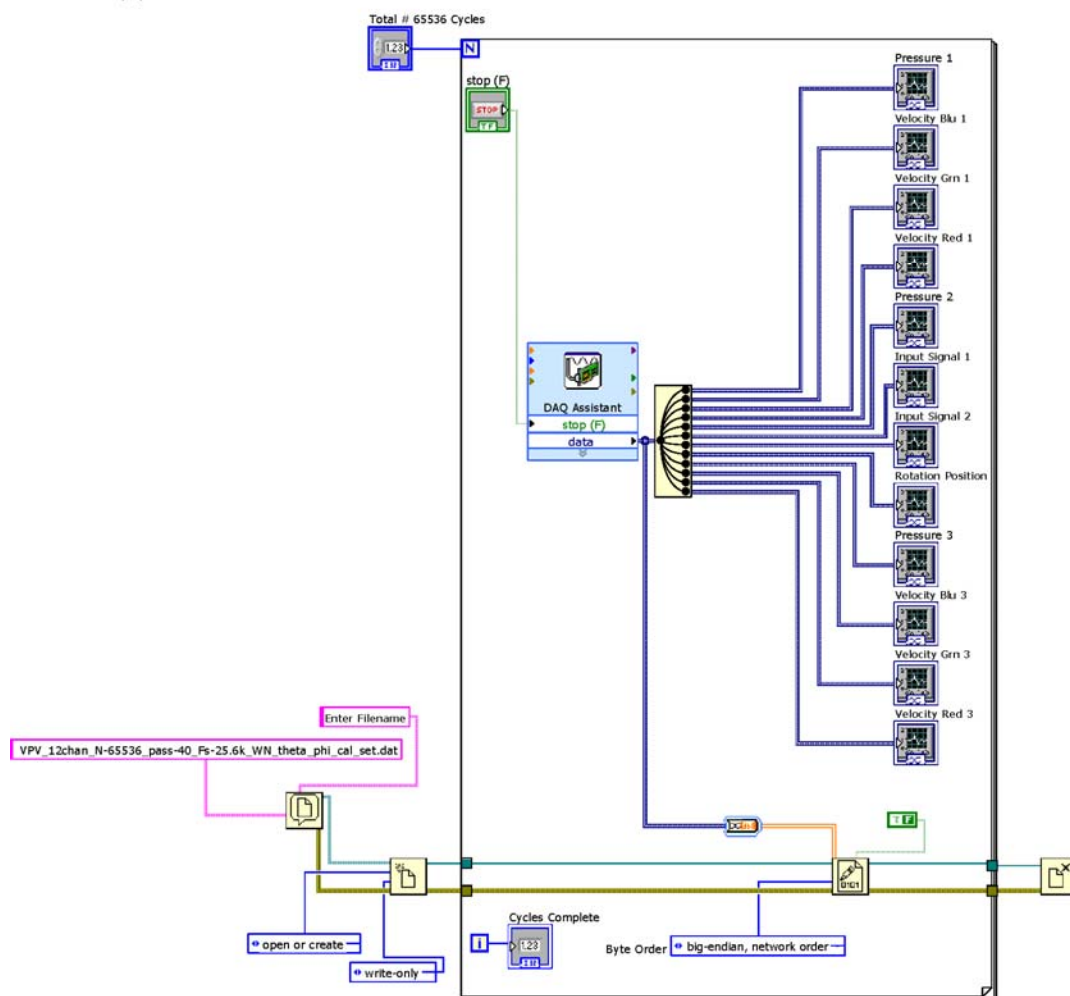
APPENDIX B. COMPUTER CODE

A. LABVIEW VI

Pressure_Velocity_12chan_bin_freq_cal_III.vi
C:\Documents and Settings\jvcaulk\Desktop\Test\Pressure_Velocity_12chan_bin_freq_cal_III.vi
Last modified on 10/15/2009 at 10:23 AM
Printed on 11/2/2009 at 10:09 AM

Page 1





B. MATLAB – POLAR_323.M

```
%Polar Plot Beampatterns of Microflow sensor
%Program to read in binary calibration data and
%compute polar plots of all elements of
%Microflow 323 sensor
%using a Microflow USP 4 channel sensor and an
%ACO Calibrated Pressure Microphone
%Naval Postgraduate School Monterey, CA
%CDR Jeff Caulk USN
%29 Oct 2009

clear all
clc

%ensure these parameters match those specified in the dataset title
maxchan = 12; %maximum number of data channels captured
numpass = 12; %number of loop iterations in vi sampling routine
N = 200000; %number of samples per channel per pass in vi binary data file
Fs = 2.56E4; %sampling freq of DAQ
cfreq=1000; % filter center freq
M = numpass * N; %total sample length per channel

bindata = zeros(M,maxchan); % initialize matrix to allocate memory

%read data file
%calibration source signal is a 1kHz output from signal generator at 1Vpp
%signal is amplified to an average of ~3Vpp then routed to driver(s)
fid = fopen('G:\Matlab\1 Sep 2009\VPV_12chan_N-200K_pass-12_Fs-25.6k_1
kHz_1Vpp_323-prs_360rot_set11.dat','r');
frewind(fid) % top of file

%Channel Designations
%DAQ Channel Designations
prs1 = 1; %sensor 323 pressure
blu1 = 2; %sensor 323 blu velocity Z axis
grn1 = 3; %sensor 323 grn velocity X axis
red1 = 4; %sensor 323 red velocity Y axis
prs2 = 5; %ACO calibrated pressure mic
src1 = 6; %Input Signal 1
src2 = 7; %Input Signal 2
psit = 8; %rotational position
prs3 = 9; %sensor 324 pressure
blu3 = 10; %sensor 324 blu velocity Z axis
grn3 = 11; %sensor 324 grn velocity X axis
```

```

red3 = 12; %sensor 324 red velocity Y axis

%2.4M datapoints = 96sec/chan - ~61.5 sec for 360° rotation
%read data from binary file in successive passes, as it was saved by vi
for j = 1:numpass
    nextcol = (j-1)*N+1; %200K datapoint increments
    bindata(nextcol:j*N,1:maxchan) = fread(fid, [N,maxchan] , 'float64', 'ieee-be');
%get 200K/chan datapoints
end
%2.4M datapoints = 96sec/chan

status = fclose('all'); %close all binary files

%Process data
bindata = bindata'; %change to row data

%plot rotator position - ensure 2x transitions for cut lines
t1 = linspace(0,M-1,M)*(1/Fs); %full time period ~96 secs
figure;plot(t1,bindata(psit,:)); %position data
title('Rotator Angular Position Voltage vs. Time')
xlabel('Time (s)')
ylabel('Voltage (V)')
%save image to disk
saveas(gcf, 'Rotator_Voltage_323.fig', 'fig');
close
clear t1; % freeup memory

%convert to frequency domain
bindataf(prs1:red1,:) = fftshift(fft(fftshift(bindata(prs1:red1,:),2),[],2),2);
%Microflow
bindataf(prs2,:) = fftshift(fft(fftshift(bindata(prs2,:)))); %prs mic
bindataf(src1,:) = fftshift(fft(fftshift(bindata(src1,:)))); %src 1
%bindataf(src2,:) = fftshift(fft(fftshift(bindata(src2,:)))); %src 2

freq=[-M/2:(M/2-1)]/(M/Fs);
plotstart = floor(M/2 + 1); %begin at freq 0
plotend = floor(plotstart + 3E3*M/Fs); %cut off higher freqs

figure %confirm source signal(s)
plot(freq(plotstart:plotend),abs(bindataf(src1,plotstart:plotend)),'b') %src1
% plot(freq(plotstart:plotend),abs(bindataf(src2,plotstart:plotend)),'r') %src2
title(['FFT Source Signal - Raw Data - ',num2str(cfreq),'Hz Signal'])
xlabel('Frequency (Hz)')
ylabel('Amplitude')
%save image to disk

```

```

saveas(gcf, 'Polar_Plot_FFT_Source_323.fig', 'fig');
close

figure %plot raw sensor data freq spectrum - sensor 323
    subplot(5,1,1)
    plot(freq(plotstart:plotend),abs(bindataf(prs2,plotstart:plotend)), 'm') %prs 2
    title(['FFT Calibrated Mic - Raw Data - Pressure ', num2str(cfreq), 'Hz Signal'])
    axis([-50 3000 0 5000])
    subplot(5,1,2)
    plot(freq(plotstart:plotend),abs(bindataf(prs1,plotstart:plotend)), 'k')
    title('FFT 323 Sensor - Raw Data - Pressure')
    axis([-50 3000 0 5000])
    subplot(5,1,3)
    plot(freq(plotstart:plotend),abs(bindataf(blu1,plotstart:plotend)), 'b')
    title('FFT 323 Sensor - Raw Data - Blue Velocity')
    axis([-50 3000 0 5000])
    subplot(5,1,4)
    plot(freq(plotstart:plotend),abs(bindataf(grn1,plotstart:plotend)), 'g')
    title('FFT 323 Sensor - Raw Data - Green Velocity')
    axis([-50 3000 0 5000])
    subplot(5,1,5)
    plot(freq(plotstart:plotend),abs(bindataf(red1,plotstart:plotend)), 'r')
    title('FFT 323 Sensor - Raw Data - Red Velocity')
    axis([-50 3000 0 5000])
%save image to disk
saveas(gcf, ['Polar_Plot_Raw_FFT_Sensors_323.fig'], 'fig');
close
clear bindataf; %freeup memory

%truncate data to 360° wide ~= 61.5 sec
%this routine looks for the first instance of a voltage crossing
%of a certain level then looks for the same crossing further down the file
%those two crossings are used to determine
%the approximate width of 360° of data signifying full rotation
%since the voltage signal is slightly oscillatory due to the bouncing of
%the potentiometer arm on the contact winding while the rotator is moving
%successive crossings occur just a few bins apart and must be ruled out
%the "bin>1000" portion performs this logic
%1.3V - voltage crossing level is close to max nominal battery voltage of 1.5V
%the lower value is used in case the battery is somewhat discharged
cutlimit = 1.3; %voltage crossing level
[psitcol] = find(bindata(psit,:) > cutlimit); %find ALL transiton points greater then
the limit
cutstart = psitcol(1); %first cut point bin index

```

```

endcol = find(diff(psitcol)>1000) + 1; %find the next relative bin index indicating
the end cut point
spread = psitcol(endcol) - cutstart; %width of cut
cutend = floor((cutstart + spread)/2)*2; %last cut bin index - ensure even integer

cutdata = bindata(:,cutstart:cutend); %data assignment - discard remainder of the
raw binary data
clear bindata; %freeup memory

%Professor Kevin Smith filter code incorporated
M2 = length(cutdata(1,:)); % ~61.5 secs
t = linspace(0,M2-1,M2)*(1/Fs);
freq = [-M2/2:(M2/2-1)]/(M2/Fs);

%baseband in time-domain
baseband = exp(i*2*pi*cfreq*t);
%de-mean data
cutdata = (cutdata - (mean(cutdata,2)*ones(1,M2))) .*
(ones(maxchan,1)*baseband);
clear baseband; %freeup memory

%apply filter and transform to freq domain
dataf = fftshift(fft(fftshift(cutdata(prs1:prs2,:),2),[],2),2); %1-4 Microflown, 5 Aco
%apply filter around 1kHz tone, reduce bandwidth to
%8192 sample points
Ns = 8192; % Use for 1K signal
df = freq(2) - freq(1);
dt = 1/(Ns*df);
nc = floor(M2/2) + 1;
dataf2 = dataf(:,nc-Ns/2:nc+Ns/2-1);
filt2 = ones(prs2,1) * hann(Ns)';
t2 = [0:dt:(Ns-1)*dt];
scale = M2/Ns;
data2 = fftshift(ifft(fftshift(dataf2 .* filt2,2),[],2),2) ./ scale;

%Plot processed FFT data
plotstart = floor(M2/2 + 1);
plotend = floor(plotstart + 5E3*M2/Fs); %cut off higher freqs

figure
    subplot(5,1,1)
    plot(freq(plotstart:plotend),abs(dataf(prs2,plotstart:plotend)), 'm') %prs 2
    title(['FFT Calibrated Mic - Processed Data - Pressure ',num2str(cfreq),'Hz
Signal'])
    axis([-50 3000 0 5000])

```

```

subplot(5,1,2)
plot(freq(plotstart:plotend),abs(dataf(prs1,plotstart:plotend)), 'k')
title('FFT 323 Sensor - Processed Data - Pressure')
axis([-50 3000 0 5000])
subplot(5,1,3)
plot(freq(plotstart:plotend),abs(dataf(blu1,plotstart:plotend)), 'b')
title('FFT 323 Sensor - Processed Data - Blue Velocity')
axis([-50 3000 0 5000])
subplot(5,1,4)
plot(freq(plotstart:plotend),abs(dataf(grn1,plotstart:plotend)), 'g')
title('FFT 323 Sensor - Processed Data - Green Velocity')
axis([-50 3000 0 5000])
subplot(5,1,5)
plot(freq(plotstart:plotend),abs(dataf(red1,plotstart:plotend)), 'r')
title('FFT 323 Sensor - Processed Data - Red Velocity')
axis([-50 3000 0 5000])
%save image to disk
saveas(gcf,'Polar_Plot_Processed_FFT_Sensors_323.fig', 'fig');
close

%Plot processed polar data before realignment
%voltage polar plots
theta = (0:(2*pi)/(Ns-1):2*pi);
figure
polar(theta,abs(data2(prs2,:)), 'm');
title(['Sensor 323 Centered Processed Data Before Realignment - ',num2str(cfreq),'Hz Signal']);
hold on
polar(theta,abs(data2(prs1,:)), 'k');
polar(theta,abs(data2(grn1,:)), 'g');
polar(theta,abs(data2(blu1,:)), 'b');
polar(theta,abs(data2(red1,:)), 'r');
legend('ACO','Prs1','Grn1','Blu1','Red1','Location','NorthEastOutside')
%save image to disk
saveas(gcf,'Polar_Plot_Processed_Polar_323_Before_Realignment.fig', 'fig');
close

%normalized plots
figure
polar(theta,abs(data2(prs2,:))/max(abs(data2(prs2,:))), 'm');
title(['Sensor 323 Centered Processed Normalized Data Before Realignment - ',num2str(cfreq),'Hz Signal']);
hold on
polar(theta,abs(data2(prs1,:))/max(abs(data2(prs1,:))), 'k');
polar(theta,abs(data2(grn1,:))/max(abs(data2(grn1,:))), 'g');

```

```

polar(theta,abs(data2(blu1,:))/max(abs(data2(blu1,:))), 'b');
polar(theta,abs(data2(red1,:))/max(abs(data2(red1,:))), 'r');
legend('ACO','Prs1','Grn1','Blu1','Red1','Location','NorthEastOutside')

%save image to disk
saveas(gcf,'Polar_Plot_Processed_Polar_323_Norm_Before_Realignment.fig',
'fig');
close

%rearrange data align to array aspect
%use minimum response of red sensor to determine 0° position
[C1,I1] = min(abs(data2(red1,1:Ns/2))); %find 0° position first half
[C2,I2] = min(abs(data2(red1,Ns/2+1:Ns))); %find 0° position second half
%finding 2 minima squares up plot better than a single index
I = floor((I1 + I2)/2); %average 0° index
%rearrange data to begin at 0° position
tempdata(:,1:Ns-I) = data2(:,I+1:Ns);
tempdata(:,Ns-I+1:Ns) = data2(:,1:I);
data2 = tempdata; %reassign rearranged data
clear tempdata; %freeup memory

theta = (0:(2*pi)/(Ns-1):2*pi);

%Plot processed polar data after realignment
%voltage plots
figure
polar(theta,abs(data2(prs2,:)), 'm');
title(['Sensor 323 Centered Processed Data After Realignment - ',num2str(cfreq),'Hz Signal']);
hold on
polar(theta,abs(data2(prs1,:)), 'k');
polar(theta,abs(data2(grn1,:)), 'g');
polar(theta,abs(data2(blu1,:)), 'b');
polar(theta,abs(data2(red1,:)), 'r');
legend('ACO','Prs1','Grn1','Blu1','Red1','Location','NorthEastOutside')
%save image to disk
saveas(gcf, 'Polar_Plot_Processed_Polar_323_After_Realignment.fig', 'fig');
close

%normalized plots
figure
polar(theta,abs(data2(prs2,:))/max(abs(data2(prs2,:))), 'm');
title(['Sensor 323 Centered Processed Normalized Data After Realignment - ',num2str(cfreq),'Hz Signal']);
hold on

```

```

polar(theta,abs(data2(prs1,:))/max(abs(data2(prs1,:))), 'k');
polar(theta,abs(data2(grn1,:))/max(abs(data2(grn1,:))), 'g');
polar(theta,abs(data2(blu1,:))/max(abs(data2(blu1,:))), 'b');
polar(theta,abs(data2(red1,:))/max(abs(data2(red1,:))), 'r');
legend('ACO','Prs1','Grn1','Blu1','Red1','Location','NorthEastOutside')
%save image to disk
saveas(gcf, 'Polar_Plot_Processed_Polar_323_Norm_After_Realignment.fig',
'fig');
close

save 323-Polar.mat %save data to disk for post-processing

```

C. MATLAB –TRANSFER_FUNCTION_CAL_323.M

```
%Transfer Functions
%Program to read in binary calibration data and compute transfer function
%between Microflow pressure sensor and velocity sensors using a Microflow 4
%channel sensor and an ACO Calibrated Pressure Microphone
%Naval Postgraduate School Monterey, CA
%CDR Jeff Caulk USN
%29 Oct 2009
%Note: This composite file produces time series and frequency domain plots to
%compare input and output signals for testing.
%The individual sensor files process the transfer functions only.
clear all
clc
%ensure these parameters match those specified in the dataset title
maxchan = 12; %maximum number of data channels captured
numpass = 40; %number of loop iterations in vi sampling routine
N = 2^16; %number of samples per channel per pass in vi binary data file
Fs = 2.56E4; %sampling freq of DAQ
M = numpass * N; %total sample length per channel
NFFT = N;
overlap = NFFT/2; %overlap window vector for computing transfer function
window = hanning(NFFT); %hanning window vector for computing transfer
function

bindata = zeros(M,maxchan); %initialize matrix to allocate memory

%DAQ Channel Designations
prs1 = 1; %sensor 323 pressure
blu1 = 2; %sensor 323 blu velocity Z axis
grn1 = 3; %sensor 323 grn velocity X axis
red1 = 4; %sensor 323 red velocity Y axis
prs2 = 5; %ACO calibrated pressure mic
src1 = 6; %Input Signal 1
src2 = 7; %Input Signal 2
psit = 8; %rotational position - DC
prs3 = 9; %sensor 324 pressure
blu3 = 10; %sensor 324 blu velocity Z axis
grn3 = 11; %sensor 324 grn velocity X axis
red3 = 12; %sensor 324 red velocity Y axis

%read data file
%calibration source signal is White Noise output from signal generator at 5Vpp
%signal is amplified to an average of ~7Vpp then routed to driver(s)
%multi frequency composition of White Noise signal precludes an absolute
```

```

%final amplification measurement
aspect = 'Theta 0° Phi 90°'; %set parameters for plot titles that correspond to
dataset
amplitude = '1';
sensor = '323';
fid = fopen('E:\4 Nov 2009\VPV_12chan_N-65536_pass-40_Fs-
25.6k_WN_0theta_90phi_Bluca1_All_set04.dat','r');
frewind(fid) % top of file

%40*2^16 datapoints = 102.4sec/chan
%read data from binary file in successive passes, as it was saved by vi
for j = 1:numpass
    nextcol = (j-1)*N+1; %2^16 datapoint increments
    bindata(nextcol:j*N,1:maxchan) = fread(fid, [N,maxchan] , 'float64', 'ieee-be');
%get 2^16/chan datapoints
end
%40*2^16 datapoints = 102.4sec/chan

status = fclose('all'); %close all binary files

%Process data
bindata = bindata'; %change to row data

t1 = linspace(0,M-1,M)*(1/Fs); %full time period ~102.4 secs
plotstart = 1; %start at beginning of time series
plotend = floor(plotstart + M/10 - 1); %cut off remainder of time series data for
display
figure %plot raw time series sensor data - sensor 323
    subplot(5,1,1)
    plot(t1(plotstart:plotend),bindata(prs2,plotstart:plotend),'m')
    title('Time Series Data - Calibrated Mic - Raw Data - Pressure')
    subplot(5,1,2)
    plot(t1(plotstart:plotend),bindata(prs1,plotstart:plotend),'k')
    title('323 Sensor - Raw Data - Pressure')
    subplot(5,1,3)
    plot(t1(plotstart:plotend),bindata(blu1,plotstart:plotend),'b')
    title('323 Sensor - Raw Data - Blue Velocity')
    subplot(5,1,4)
    plot(t1(plotstart:plotend),bindata(grn1,plotstart:plotend),'g')
    title('323 Sensor - Raw Data - Green Velocity')
    subplot(5,1,5)
    plot(t1(plotstart:plotend),bindata(red1,plotstart:plotend),'r')
    title('323 Sensor - Raw Data - Red Velocity')
clear t1;

```

```

%transfer source signals to freq domain
bindataf(src1:src2,:) = fftshift(fft(fftshift(bindata(src1:src2,:),2),[],2),2);

freq=[-M/2:(M/2-1)]/(M/Fs);
plotstart = M/2 + 1; %begin at freq 0 for display
plotend = floor(plotstart + 5E3*M/Fs); %cut off higher freqs for display

figure %confirm source signal(s) are correct freqs
    plot(freq(plotstart:plotend),abs(bindataf(src1,plotstart:plotend)),'b')
%    plot(freq(plotstart:plotend),abs(bindataf(src2,plotstart:plotend)),'r')
    title(['FFT Source Signal for ',sensor,' Sensor at ',aspect,' Incidence - Raw
Data'])
    xlabel('Frequency (Hz)')
    ylabel('Amplitude')
    axis([-50 4000 0 500])
clear bindataf; %freeup memory

%transfer sensor signals to freq domain
bindataf(prs1:prs2,:) = fftshift(fft(fftshift(bindata(prs1:prs2,:),2),[],2),2);

sensor = '323';
figure %plot raw sensor data freq spectrum - sensor 323
    subplot(5,1,1)
    plot(freq(plotstart:plotend),abs(bindataf(prs2,plotstart:plotend)),'m')
    title(['FFT Calibrated Mic at ',aspect,' Incidence - Raw Data - Pressure'])
    axis([-50 4000 0 100])
    subplot(5,1,2)
    plot(freq(plotstart:plotend),abs(bindataf(prs1,plotstart:plotend)),'k')
    title(['FFT ',sensor,' Sensor at ',aspect,' Incidence - Raw Data - Pressure'])
    axis([-50 4000 0 100])
    subplot(5,1,3)
    plot(freq(plotstart:plotend),abs(bindataf(blu1,plotstart:plotend)),'b')
    title(['FFT ',sensor,' Sensor at ',aspect,' Incidence - Raw Data - Blue Velocity'])
    axis([-50 4000 0 100])
    subplot(5,1,4)
    plot(freq(plotstart:plotend),abs(bindataf(grn1,plotstart:plotend)),'g')
    title(['FFT ',sensor,' Sensor at ',aspect,' Incidence - Raw Data - Green
Velocity'])
    axis([-50 4000 0 100])
    subplot(5,1,5)
    plot(freq(plotstart:plotend),abs(bindataf(red1,plotstart:plotend)),'r')
    title(['FFT ',sensor,' Sensor at ',aspect,' Incidence - Raw Data - Red Velocity'])
    axis([-50 4000 0 100])

clear bindataf; %freeup memory

```

```
% Compute transfer function between pressure mic and pressure 323
gammap1 = mscohere(bindata(prs2,:),bindata(prs1,:),window,overlap,NFFT,Fs);
%Magnitude Squared Coherence Estimate
[Pp1p,F] = cpsd(bindata(prs2,:),bindata(prs1,:),window,overlap,NFFT,Fs);
%Cross Power Spectral Density - Welch's method
[Pp2,F] = pwelch(bindata(prs2,:),window,overlap,NFFT,Fs); %Power Spectral
Density - Welch's method
```

```
Hp1 = Pp2./Pp1p;
senselement = 'Pressure';
```

```
figure
subplot(3,1,1)
plot(F,gammap1,'k')
ylabel('Coherence')
title(['Coherence - Calibrated Pressure Sensor and ',sensor,' Sensor ',
senselement,' ',amplitude,'Vpp ',aspect,'° Incidence'])
axis([-50 3000 0 1.1])
subplot(3,1,2)
plot(F,abs(Hp1),'k')
ylabel('Transfer Function Amplitude')
title(['Amplitude of Transfer Function - Calibrated Pressure Sensor and ',sensor,'
Sensor ',senselement,' ',amplitude,'Vpp ',aspect,'° Incidence'])
axis([-50 3000 0 1.1])
subplot(3,1,3)
plot(F, rad2deg(angle(Hp1)), 'k')
title(['Phase Difference - Calibrated Pressure Sensor and ',sensor,' Sensor ',
senselement,' ',amplitude,'Vpp ',aspect,'° Incidence'])
ylabel('Phase Difference (deg)')
xlabel('Frequency (Hz)')
axis([-50 3000 -200 200])
```

```
% Compute transfer function between pressure 323 and blue velocity 323
gammap1 =
mscohere(bindata(prs1,:),bindata(blu1,:),window,overlap,NFFT,Fs);%Magnitude
Squared Coherence Estimate
[Pb1p,F] = cpsd(bindata(prs1,:),bindata(blu1,:),window,overlap,NFFT,Fs);
%Cross Power Spectral Density - Welch's method
[Pp1,F] = pwelch(bindata(prs1,:),window,overlap,NFFT,Fs); %Power Spectral
Density - Welch's method
```

```
Hb1 = Pp1./Pb1p;
senselement = 'Blue Velocity';
```

```

figure
subplot(3,1,1)
plot(F,gammab1,'b')
ylabel('Coherence')
title(['Coherence - ',sensor,' Pressure Sensor and ',sensor,' Sensor ',senselement,' ',amplitude,'Vpp ',aspect,'° Incidence'])
axis([-50 3000 0 1.1])
subplot(3,1,2)
plot(F,abs(Hb1),'b')
ylabel('Transfer Function Amplitude')
title(['Amplitude of Transfer Function - ',sensor,' Pressure Sensor and ',sensor,' Sensor ',senselement,' ',amplitude,'Vpp ',aspect,'° Incidence'])
axis([-50 3000 0 8.1])
subplot(3,1,3)
plot(F, rad2deg(angle(Hb1)),'b')
title(['Phase Difference - ',sensor,' Pressure Sensor and ',sensor,' Sensor ',senselement,' ',amplitude,'Vpp ',aspect,'° Incidence'])
ylabel('Phase Difference (deg)')
xlabel('Frequency (Hz)')
axis([-50 3000 -200 200])

% Compute transfer function between pressure 323 and red velocity 323
gammar1 =
mscohere(bindata(prs1,:),bindata(red1,:),window,overlap,NFFT,Fs);%Magnitude Squared Coherence Estimate
[Pr1p,F] = cpsd(bindata(prs1,:),bindata(red1,:),window,overlap,NFFT,Fs);
%Cross Power Spectral Density - Welch's method
[Pp1,F] = pwelch(bindata(prs1,:),window,overlap,NFFT,Fs); %Power Spectral Density - Welch's method

Hr1 = Pp1./Pr1p;
senselement = 'Red Velocity';

figure
subplot(3,1,1)
plot(F,gammar1,'r')
ylabel('Coherence')
title(['Coherence - ',sensor,' Pressure Sensor and ',sensor,' Sensor ',senselement,' ',amplitude,'Vpp ',aspect,'° Incidence'])
axis([-50 3000 0 1.1])
subplot(3,1,2)
plot(F,abs(Hr1),'r')
ylabel('Transfer Function Amplitude')
title(['Amplitude of Transfer Function - ',sensor,' Pressure Sensor and ',sensor,' Sensor ',senselement,' ',amplitude,'Vpp ',aspect,'° Incidence'])

```

```

axis([-50 3000 0 1.1])
subplot(3,1,3)
plot(F, rad2deg(angle(Hr1)), 'r')
title(['Phase Difference - ', sensor, ' Pressure Sensor and ', sensor, ' Sensor',
'senselement, ', amplitude, 'Vpp ', aspect, '° Incidence'])
ylabel('Phase Difference (deg)')
xlabel('Frequency (Hz)')
axis([-50 3000 -200 200])

% Compute transfer function between pressure 323 and green velocity 323
gammag1 =
mscohere(bindata(prs1,:), bindata(grn1,:), window, overlap, NFFT, Fs); %Magnitude
Squared Coherence Estimate
[Pg1p, F] = cpsd(bindata(prs1,:), bindata(grn1,:), window, overlap, NFFT, Fs);
%Cross Power Spectral Density - Welch's method
[Pp1, F] = pwelch(bindata(prs1,:), window, overlap, NFFT, Fs); %Power Spectral
Density - Welch's method

Hg1 = Pp1./Pg1p;
senselement = 'Green Velocity';

figure
subplot(3,1,1)
plot(F, gammag1, 'g')
ylabel('Coherence')
title(['Coherence - ', sensor, ' Pressure Sensor and ', sensor, ' Sensor',
'senselement, ', amplitude, 'Vpp ', aspect, '° Incidence'])
axis([-50 3000 0 1.1])
subplot(3,1,2)
plot(F, abs(Hg1), 'g')
ylabel('Transfer Function Amplitude')
title(['Amplitude of Transfer Function - ', sensor, ' Pressure Sensor and ', sensor, ' Sensor',
'senselement, ', amplitude, 'Vpp ', aspect, '° Incidence'])
axis([-50 3000 0 5])
subplot(3,1,3)
plot(F, rad2deg(angle(Hg1)), 'g')
title(['Phase Difference - ', sensor, ' Pressure Sensor and ', sensor, ' Sensor',
'senselement, ', amplitude, 'Vpp ', aspect, '° Incidence'])
ylabel('Phase Difference (deg)')
xlabel('Frequency (Hz)')
axis([-50 3000 -200 200])

```

D. MATLAB –TRANSFER_FUNCTION_CAL_323_GRN_MULTI.M

```
%323 Green and Pressure Transfer Functions
%Program to read in binary calibration data
%from multiple data files and
%compute transfer function between
%Microflown pressure sensor and green velocity sensor
%and Microflown pressure sensor and Aco mic
%using a Microflown 4 channel sensor and an
%ACO Calibrated Pressure Microphone
%Naval Postgraduate School Monterey, CA
%CDR Jeff Caulk USN
%29 Oct 2009

%Note: This individual sensor file processes multiple datafiles
%and computes the transfer functions for the prs and grn sensors only.
%The composite file produces time series and frequency domain plots to
%compare input and output signals for testing. clear all
clear all
clc

%ensure these parameters match those specified in the dataset title
maxchan = 12; %maximum number of data channels captured
numpass = 40; %number of loop iterations in vi sampling routine
N = 2^16; %number of samples per channel per pass in vi binary data file
Fs = 2.56E4; %sampling freq of DAQ
M = numpass * N; %total sample length per channel
NFFT = N;
overlap = NFFT/2; %overlap window vector for computing transfer function
window = hanning(NFFT); %hanning window vector for computing transfer
function

bindata = zeros(M,maxchan); %initialize matrix to allocate memory

%DAQ Channel Designations in vi
prs1 = 1; % sensor 323 pressure
blu1 = 2; % sensor 323 blu velocity Z axis
grn1 = 3; % sensor 323 grn velocity X axis
red1 = 4; % sensor 323 red velocity Y axis
prs2 = 5; % ACO calibrated pressure mic
src1 = 6; % Input Signal 1
src2 = 7; % Input Signal 2
psit = 8; % rotational position - DC
prs3 = 9; % sensor 324 pressure
blu3 = 10; % sensor 324 blu velocity Z axis
```

```

grn3 = 11; % sensor 324 grn velocity X axis
red3 = 12; % sensor 324 red velocity Y axis

numdatasets = 3; %number of complete data sets

for q = 1:numdatasets
    % read data file
    %calibration source signal is White Noise output from signal generator at 5Vpp
    %signal is amplified to an average of ~7Vpp then routed to driver(s)
    %multi frequency composition of White Noise signal precludes an absolute
    %final amplification measurement
    aspect = 'Theta 90° Phi 0°'; %set parameters for plot titles that correspond to
dataset
    amplitude = '5';
    sensor = '323';

    %dataset file names
    if q == 1
        fid = fopen('E:\4 Nov 2009\VPV_12chan_N-65536_pass-40_Fs-
25.6k_WN_90theta_0phi_Grncal_All_set07.dat','r');
    elseif q == 2
        fid = fopen('E:\4 Nov 2009\VPV_12chan_N-65536_pass-40_Fs-
25.6k_WN_90theta_0phi_Grncal_All_set08.dat','r');
    elseif q == 3
        fid = fopen('E:\4 Nov 2009\VPV_12chan_N-65536_pass-40_Fs-
25.6k_WN_90theta_0phi_Grncal_All_set09.dat','r');
    end

    frewind(fid) %top of file

    %40*2^16 datapoints = 102.4sec/chan
    %read data from binary file in successive passes, as it was saved by vi
    for j = 1:numpass
        nextcol = (j-1)*N+1; %2^16 datapoint increments
        bindata(nextcol:j*N,1:maxchan) = fread(fid, [N,maxchan] , 'float64', 'ieee-
be'); %get 2^16/chan datapoints
    end
    %40*2^16 datapoints = 102.4sec/chan

    status = fclose('all'); %close all binary files

    %Process data
    bindata = bindata'; %change to row data

    %USE GREEN Pressure Transfer Function for Beam Former

```

```

%Compute transfer function between pressure mic and pressure 323
gammap1(:,q) =
mscohere(bindata(prs2,:),bindata(prs1,:),window,overlap,NFFT,Fs); %Magnitude
Squared Coherence Estimate
[Pp1p(:,q),F] = cpsd(bindata(prs2,:),bindata(prs1,:),window,overlap,NFFT,Fs);
%Cross Power Spectral Density - Welch's method
[Pp2(:,q),F] = pwelch(bindata(prs2,:),window,overlap,NFFT,Fs); %Power
Spectral Density - Welch's method

Hp1(:,q) = Pp2(:,q)./Pp1p(:,q);

%Compute transfer function between pressure 323 and green velocity 323
gammap1(:,q) =
mscohere(bindata(prs1,:),bindata(grn1,:),window,overlap,NFFT,Fs); %Magnitude
Squared Coherence Estimate
[Pg1p(:,q),F] = cpsd(bindata(prs1,:),bindata(grn1,:),window,overlap,NFFT,Fs);
%Cross Power Spectral Density - Welch's method
[Pp1(:,q),F] = pwelch(bindata(prs1,:),window,overlap,NFFT,Fs); %Power
Spectral Density - Welch's method

Hg1(:,q) = Pp1(:,q)./Pg1p(:,q);

clear bindata;
end
%end data read and computation loop

%Display all computed functions for comparison
%to ensure data agrees prior to averaging
%USE GREEN Pressure Transfer Function for Beam Former
senselement = 'Pressure';
figure
subplot(3,1,1)
hold on
plot(F,gammap1(:,2),'m')
plot(F,gammap1(:,3),'c')
plot(F,gammap1(:,1),'k')
ylabel('Coherence')
title(['Coherence - Calibrated Pressure Sensor and ',sensor,' Sensor
',senselement,' ',amplitude,'Vpp ',aspect,' Incidence'])
axis([-50 3000 0 1.1])
subplot(3,1,2)
hold on
plot(F,abs(Hp1(:,2)),'m')
plot(F,abs(Hp1(:,3)),'c')
plot(F,abs(Hp1(:,1)),'k')

```

```

ylabel('Transfer Function Amplitude')
title(['Amplitude of Transfer Function - Calibrated Pressure Sensor and ',sensor,'
Sensor ',senselement,' ',amplitude,'Vpp ',aspect,' Incidence'])
axis([-50 3000 0 5])
subplot(3,1,3)
hold on
plot(F, rad2deg(angle(Hp1(:,2))), 'm')
plot(F, rad2deg(angle(Hp1(:,3))), 'c')
plot(F, rad2deg(angle(Hp1(:,1))), 'k')
title(['Phase Difference - Calibrated Pressure Sensor and ',sensor,' Sensor
',senselement,' ',amplitude,'Vpp ',aspect,' Incidence'])
ylabel('Phase Difference (deg)')
xlabel('Frequency (Hz)')
axis([-50 3000 -200 200])

%save image to disk
saveas(gcf, [sensor,'_Prs-Grn_Transfer.fig'], 'fig');
close

senselement = 'Green Velocity';
figure
subplot(3,1,1)
hold on
plot(F,gammag1(:,2),'m')
plot(F,gammag1(:,3),'c')
plot(F,gammag1(:,1),'g')
ylabel('Coherence')
title(['Coherence - ',sensor,' Pressure Sensor and ',sensor,' Sensor
',senselement,' ',amplitude,'Vpp ',aspect,' Incidence'])
axis([-50 3000 0 1.1])
subplot(3,1,2)
hold on
plot(F,abs(Hg1(:,2)), 'm')
plot(F,abs(Hg1(:,3)), 'c')
plot(F,abs(Hg1(:,1)), 'g')
ylabel('Transfer Function Amplitude')
title(['Amplitude of Transfer Function - ',sensor,' Pressure Sensor and ',sensor,'
Sensor ',senselement,' ',amplitude,'Vpp ',aspect,' Incidence'])
axis([-50 3000 0 15])
subplot(3,1,3)
hold on
plot(F, rad2deg(angle(Hg1(:,2))), 'm')
plot(F, rad2deg(angle(Hg1(:,3))), 'c')
plot(F, rad2deg(angle(Hg1(:,1))), 'g')

```

```

title(['Phase Difference - ',sensor,' Pressure Sensor and ',sensor,' Sensor
',senselement,' ',amplitude,'Vpp ',aspect,' Incidence'])
ylabel('Phase Difference (deg)')
xlabel('Frequency (Hz)')
axis([-50 3000 -200 200])

%save image to disk
saveas(gcf, [sensor,'_Grn_Transfer.fig'], 'fig');
close

save 323-Grn-Prs.mat Hg1 Hp1 %save transfer function to disk for post-
processing
save 323-Grn-Prs_All.mat %save data to disk for post-processing

```

E. MATLAB –TRANSFER_FUNCTION_CAL_323_PRS_MULTI.M

```
%323 Pressure Transfer Functions
%Program to read in binary calibration data
%from multiple data files and
%compute transfer function between
%Microflown pressure sensor and Aco mic
%in both Red MRA and Green MRA orientations
%using a Microflown 4 channel sensor and an
%ACO Calibrated Pressure Microphone
%Naval Postgraduate School Monterey, CA
%CDR Jeff Caulk USN
%29 Oct 2009

%Note: This individual sensor file processes multiple datafiles
%and computes the transfer functions for the prs sensors only.
%The composite file produces time series and frequency domain plots to
%compare input and output signals for testing. clear allclear all
clear all
clc

%ensure these parameters match those specified in the dataset title
maxchan = 12; %maximum number of data channels captured
numpass = 40; %number of loop iterations in vi sampling routine
N = 2^16; %number of samples per channel per pass in vi binary data file
Fs = 2.56E4; %sampling freq of DAQ
M = numpass * N; %total sample length per channel
NFFT = N;
overlap = NFFT/2; %overlap window vector for computing transfer function
window = hanning(NFFT); %hanning window vector for computing transfer
function

bindata = zeros(M,maxchan); %initialize matrix to allocate memory

%DAQ Channel Designations in vi
prs1 = 1; % sensor 323 pressure
blu1 = 2; % sensor 323 blu velocity Z axis
grn1 = 3; % sensor 323 grn velocity X axis
red1 = 4; % sensor 323 red velocity Y axis
prs2 = 5; % ACO calibrated pressure mic
src1 = 6; % Input Signal 1
src2 = 7; % Input Signal 2
psit = 8; % rotational position - DC
prs3 = 9; % sensor 324 pressure
blu3 = 10; % sensor 324 blu velocity Z axis
```

```

grn3 = 11; % sensor 324 grn velocity X axis
red3 = 12; % sensor 324 red velocity Y axis

numdatasets = 6; %number of complete data sets

for q = 1:numdatasets
    % read data file
    %calibration source signal is White Noise output from signal generator at 5Vpp
    %signal is amplified to an average of ~7Vpp then routed to driver(s)
    %multi frequency composition of White Noise signal precludes an absolute
    %final amplification measurement
    aspect = 'Theta 90° Phi 0° and 90°'; %set parameters for plot titles that
    correspond to dataset
    amplitude = '5';
    sensor = '323';

    %dataset file names
    if q == 1
        fid = fopen('E:\4 Nov 2009\VPV_12chan_N-65536_pass-40_Fs-
25.6k_WN_90theta_0phi_Grncal_All_set07.dat','r');
    elseif q == 2
        fid = fopen('E:\4 Nov 2009\VPV_12chan_N-65536_pass-40_Fs-
25.6k_WN_90theta_0phi_Grncal_All_set08.dat','r');
    elseif q == 3
        fid = fopen('E:\4 Nov 2009\VPV_12chan_N-65536_pass-40_Fs-
25.6k_WN_90theta_0phi_Grncal_All_set09.dat','r');
    elseif q == 4
        fid = fopen('E:\4 Nov 2009\VPV_12chan_N-65536_pass-40_Fs-
25.6k_WN_90theta_90phi_Redcal_All_set01.dat','r');
    elseif q == 5
        fid = fopen('E:\4 Nov 2009\VPV_12chan_N-65536_pass-40_Fs-
25.6k_WN_90theta_90phi_Redcal_All_set02.dat','r');
    elseif q == 6
        fid = fopen('E:\4 Nov 2009\VPV_12chan_N-65536_pass-40_Fs-
25.6k_WN_90theta_90phi_Redcal_All_set03.dat','r');
    end

    frewind(fid) %top of file

    %40*2^16 datapoints = 102.4sec/chan
    %read data from binary file in successive passes, as it was saved by vi
    for j = 1:numpass
        nextcol = (j-1)*N+1; %2^16 datapoint increments
        bindata(nextcol:j*N,1:maxchan) = fread(fid, [N,maxchan] , 'float64', 'ieee-
be'); %get 2^16/chan datapoints
    end
end

```

```

end
%40*2^16 datapoints = 102.4sec/chan

status = fclose('all'); %close all BINARY files

%Process data
bindata = bindata'; %change to row data

%USE GREEN Pressure Transfer Function for Beam Former
%Compute transfer function between pressure mic and pressure 323
gammap1(:,q) =
mscohere(bindata(prs2,:),bindata(prs1,:),window,overlap,NFFT,Fs); %Magnitude
Squared Coherence Estimate
[Pp1p(:,q),F] = cpsd(bindata(prs2,:),bindata(prs1,:),window,overlap,NFFT,Fs);
%Cross Power Spectral Density - Welch's method
[Pp2(:,q),F] = pwelch(bindata(prs2,:),window,overlap,NFFT,Fs); %Power
Spectral Density - Welch's method

Hp1(:,q) = Pp2(:,q)./Pp1p(:,q);

clear bindata;
end
%end data read and computation loop

%Display all computed functions for comparison
%to ensure data agrees prior to averaging
%USE GREEN Pressure Transfer Function for Beam Former
senselement = 'Pressure';
figure
subplot(3,1,1)
hold on
plot(F,gammap1(:,2),'m')
plot(F,gammap1(:,3),'y')
plot(F,gammap1(:,1),'k')%primary grn orientation trace
plot(F,gammap1(:,5),'g')
plot(F,gammap1(:,6),'b')
plot(F,gammap1(:,4),'c')%primary red orientation trace
ylabel('Coherence')
title(['Coherence - Calibrated Pressure Sensor and ',sensor,' Sensor
',senselement,' ',amplitude,'Vpp ',aspect,' Incidence'])
axis([-5 3000 0 1.1])
subplot(3,1,2)
hold on
plot(F,abs(Hp1(:,2)),'m')
plot(F,abs(Hp1(:,3)),'y')

```

```

plot(F,abs(Hp1(:,1)), 'k')%primary grn orientation trace
plot(F,abs(Hp1(:,5)), 'g')
plot(F,abs(Hp1(:,6)), 'b')
plot(F,abs(Hp1(:,4)), 'c')%primary red orientation trace
ylabel('Transfer Function Amplitude')
title(['Amplitude of Transfer Function - Calibrated Pressure Sensor and ',sensor,'
Sensor ',senselement,' ',amplitude,'Vpp ',aspect,' Incidence'])
axis([-5 3000 0 5])
subplot(3,1,3)
hold on
plot(F, rad2deg(angle(Hp1(:,2))), 'm')
plot(F, rad2deg(angle(Hp1(:,3))), 'y')
plot(F, rad2deg(angle(Hp1(:,1))), 'k')%primary grn orientation trace
plot(F, rad2deg(angle(Hp1(:,5))), 'g')
plot(F, rad2deg(angle(Hp1(:,6))), 'b')
plot(F, rad2deg(angle(Hp1(:,4))), 'c')%primary red orientation trace
title(['Phase Difference - Calibrated Pressure Sensor and ',sensor,' Sensor
',senselement,' ',amplitude,'Vpp ',aspect,' Incidence'])
ylabel('Phase Difference (deg)')
xlabel('Frequency (Hz)')
axis([-5 3000 -200 200])

%save image to disk
saveas(gcf, [sensor,'_Prs-All_Transfer.fig'], 'fig');
close

save 323-Prs.mat Hp1 %save transfer function to disk for post-processing
save 323-Prs_All.mat %save data to disk for post-processing

```

F. MATLAB – AVERAGE_TRANSFER_FUNCTIONS.M

```
%Average and Convert Microflown sensor transfer functions
%for use in a beamformer for Hybrid Array using
%Two Microflown 4 channel sensors and an
%ACO Calibrated Pressure Microphone
%Naval Postgraduate School Monterey, CA
%CDR Jeff Caulk USN
%29 Oct 2009

clear all
clc

N = 2^15; %final length of transfer functions

%get individual transfer functions
%transfer functions are kept in individual arrays to ease code tracing
%load Sensor_Transfer_Functions_raw.mat %length 2^15+
load 323-Blu.mat %Hb1 length 2^15+1
load 323-Grn-Prs.mat %Hp1 Hg1 length 2^15+1
load 323-Red.mat %Hr1 length 2^15+1
load 324-Blu.mat %Hb3 length 2^15+1
load 324-Grn-Prs.mat %Hp3 Hg3 F length 2^15+1
load 324-Red.mat %Hr3 length 2^15+1

F = F(1:N);

%truncate transfer functions to length
%323 Sensor
Hp1 = Hp1(1:N,:);
Hb1 = Hb1(1:N,:);
Hg1 = Hg1(1:N,:);
Hr1 = Hr1(1:N,:);

%324 Sensor
Hp3 = Hp3(1:N,:);
Hb3 = Hb3(1:N,:);
Hg3 = Hg3(1:N,:);
Hr3 = Hr3(1:N,:);

%average the three transfer functions for each element
%into a single function length 1 x N
%323 Sensor
Hp1 = sum(Hp1,2)/3;
Hb1 = sum(Hb1,2)/3;
```

```

Hg1 = sum(Hg1,2)/3;
Hr1 = sum(Hr1,2)/3;

%324 Sensor
Hp3 = sum(Hp3,2)/3;
Hb3 = sum(Hb3,2)/3;
Hg3 = sum(Hg3,2)/3;
Hr3 = sum(Hr3,2)/3;

%final calculation to relate velocity sensors to Aco
%multiplication performed here to save processing time in beam former
%323 Sensor
Hb1 = Hb1 .* Hp1;
Hg1 = Hg1 .* Hp1;
Hr1 = Hr1 .* Hp1;

%324 Sensor
Hb3 = Hb3 .* Hp3;
Hg3 = Hg3 .* Hp3;
Hr3 = Hr3 .* Hp3;

sensor = '323';
senselement = 'Pressure';
figure
subplot(2,1,1)
plot(F,abs(Hp1),'k')
ylabel('Final Transfer Function Amplitude')
title(['Amplitude of Final Transfer Function - Aco Pressure Sensor and ',sensor,'
Sensor ',senselement])
axis([300 2000 0 10])
subplot(2,1,2)
plot(F, rad2deg(angle(Hp1)), 'k')
title(['Final Phase Difference - Aco Pressure Sensor and ',sensor,' Sensor
',senselement])
ylabel('Final Phase Difference (deg)')
xlabel('Frequency (Hz)')
axis([300 2000 -200 200])

%save image to disk
saveas(gcf, [sensor,'_Prs_Transfer_Final.fig'], 'fig');
close

senselement = 'Blue Velocity';
figure

```

```

subplot(2,1,1)
plot(F,abs(Hb1),'b')
ylabel('Final Transfer Function Amplitude')
title(['Amplitude of Final Transfer Function - Aco Pressure Sensor and ',sensor,'
Sensor ',senselement])
axis([300 2000 0 10])
subplot(2,1,2)
plot(F, rad2deg(angle(Hb1)),'b')
title(['Final Phase Difference - Aco Pressure Sensor and ',sensor,' Sensor
',senselement])
ylabel('Final Phase Difference (deg)')
xlabel('Frequency (Hz)')
axis([300 2000 -200 200])

```

```

%save image to disk
saveas(gcf, [sensor,'_Blu_Transfer_Final.fig'], 'fig');
close

```

```

senselement = 'Green Velocity';
figure
subplot(2,1,1)
plot(F,abs(Hg1),'g')
ylabel('Final Transfer Function Amplitude')
title(['Amplitude of Final Transfer Function - Aco Pressure Sensor and ',sensor,'
Sensor ',senselement])
axis([300 2000 0 10])
subplot(2,1,2)
plot(F, rad2deg(angle(Hg1)),'g')
title(['Final Phase Difference - Aco Pressure Sensor and ',sensor,' Sensor
',senselement])
ylabel('Final Phase Difference (deg)')
xlabel('Frequency (Hz)')
axis([300 2000 -200 200])

```

```

%save image to disk
saveas(gcf, [sensor,'_Grn_Transfer_Final.fig'], 'fig');
close

```

```

senselement = 'Red Velocity';
figure
subplot(2,1,1)
plot(F,abs(Hr1),'r')
ylabel('Final Transfer Function Amplitude')

```

```

title(['Amplitude of Final Transfer Function - Aco Pressure Sensor and ',sensor,'
Sensor ',senselement])
axis([300 2000 0 10])
subplot(2,1,2)
plot(F, rad2deg(angle(Hr1)), 'r')
title(['Final Phase Difference - Aco Pressure Sensor and ',sensor,' Sensor
',senselement])
ylabel('Final Phase Difference (deg)')
xlabel('Frequency (Hz)')
axis([300 2000 -200 200])

%save image to disk
saveas(gcf, [sensor,'_Red_Transfer_Final.fig'], 'fig');
close

```

```

sensor = '324';
senselement = 'Pressure';
figure
subplot(2,1,1)
plot(F,abs(Hp3),'k')
ylabel('Final Transfer Function Amplitude')
title(['Amplitude of Final Transfer Function - Aco Pressure Sensor and ',sensor,'
Sensor ',senselement])
axis([300 2000 0 10])
subplot(2,1,2)
plot(F, rad2deg(angle(Hp3)), 'k')
title(['Final Phase Difference - Aco Pressure Sensor and ',sensor,' Sensor
',senselement])
ylabel('Final Phase Difference (deg)')
xlabel('Frequency (Hz)')
axis([300 2000 -200 200])

%save image to disk
saveas(gcf, [sensor,'_Prs_Transfer_Final.fig'], 'fig');
close

```

```

senselement = 'Blue Velocity';
figure
subplot(2,1,1)
plot(F,abs(Hb3),'b')
ylabel('Final Transfer Function Amplitude')
title(['Amplitude of Final Transfer Function - Aco Pressure Sensor and ',sensor,'
Sensor ',senselement])

```

```

axis([300 2000 0 10])
subplot(2,1,2)
plot(F, rad2deg(angle(Hb3)), 'b')
title(['Final Phase Difference - Aco Pressure Sensor and ', sensor, ' Sensor', senselement])
ylabel('Final Phase Difference (deg)')
xlabel('Frequency (Hz)')
axis([300 2000 -200 200])

```

```

%save image to disk
saveas(gcf, [sensor, '_Blu_Transfer_Final.fig'], 'fig');
close

```

```

senselement = 'Green Velocity';
figure
subplot(2,1,1)
plot(F, abs(Hg3), 'g')
ylabel('Final Transfer Function Amplitude')
title(['Amplitude of Final Transfer Function - Aco Pressure Sensor and ', sensor, ' Sensor', senselement])
axis([300 2000 0 10])
subplot(2,1,2)
plot(F, rad2deg(angle(Hg3)), 'g')
title(['Final Phase Difference - Aco Pressure Sensor and ', sensor, ' Sensor', senselement])
ylabel('Final Phase Difference (deg)')
xlabel('Frequency (Hz)')
axis([300 2000 -200 200])

```

```

%save image to disk
saveas(gcf, [sensor, '_Grn_Transfer_Final.fig'], 'fig');
close

```

```

senselement = 'Red Velocity';
figure
subplot(2,1,1)
plot(F, abs(Hr3), 'r')
ylabel('Final Transfer Function Amplitude')
title(['Amplitude of Final Transfer Function - Aco Pressure Sensor and ', sensor, ' Sensor', senselement])
axis([300 2000 0 10])
subplot(2,1,2)
plot(F, rad2deg(angle(Hr3)), 'r')

```

```

title(['Final Phase Difference - Aco Pressure Sensor and ',sensor,' Sensor',
'senselement'])
ylabel('Final Phase Difference (deg)')
xlabel('Frequency (Hz)')
axis([300 2000 -200 200])

```

```

%save image to disk
saveas(gcf, [sensor,'_Red_Transfer_Final.fig'], 'fig');
close

```

```

%Transpose transfer functions to row format for beamformer
%323 Sensor
Hp1 = Hp1.'; %non-conjugate transpose
Hb1 = Hb1.'; %non-conjugate transpose
Hg1 = Hg1.'; %non-conjugate transpose
Hr1 = Hr1.'; %non-conjugate transpose

```

```

%324 Sensor
Hp3 = Hp3.'; %non-conjugate transpose
Hb3 = Hb3.'; %non-conjugate transpose
Hg3 = Hg3.'; %non-conjugate transpose
Hr3 = Hr3.'; %non-conjugate transpose

```

```

save Sensor_Transfer_Functions.mat Hp3 Hb3 Hg3 Hr3 Hp1 Hb1 Hg1 Hr1
save Sensor_Transfer_Functions_All.mat

```

G. MATLAB – MICROFLOWNARRAY_12CHAN_BEAMFORMER.M

```
%Beamformer for Hybrid Array using
%Two Microflown 4 channel sensors and an
%ACO Calibrated Pressure Microphone
%Naval Postgraduate School Monterey, CA
%CDR Jeff Caulk USN
%2 Dec 2009

clear all
clc

%%Setup%%
%%enter parameters to match datafile and frequency region of interest%
N = 2^15; %signal, transfer function and FFT length
numpass = 1; %30 number of N data chunks in binary file saved by vi
binN = 2^16; %binary data chunk size - must match data block size as saved in
Labview vi
Fs = 25.6E3; %DAQ 25.6kHz sampling frequency
Fo = 1000; %center frequency of interest
thetainc = 1; %angle increment degrees
phiinc = 1; %angle increment degrees
numsens = 3; %total number of sensors
numel = 4; %number of elements per sensor
%%end parameter setup - enter datafile parameters below%%

%DAQ channel assignments same as vi
prs1 = 1; %sensor 323 pressure
blu1 = 2; %sensor 323 blu velocity Z axis
grn1 = 3; %sensor 323 grn velocity X axis
red1 = 4; %sensor 323 red velocity Y axis
prs2 = 5; %ACO calibrated pressure mic
blu2 = 6; %ACO calibrated pressure no velocity - dummy
grn2 = 7; %ACO calibrated pressure no velocity - dummy
red2 = 8; %ACO calibrated pressure no velocity - dummy
prs3 = 9; %sensor 324 pressure
blu3 = 10; %sensor 324 blu velocity Z axis
grn3 = 11; %sensor 324 grn velocity X axis
red3 = 12; %sensor 324 red velocity Y axis

%read data file
%%set parameters for plot titles that correspond to dataset%%
%Hybrid
clims = [77.4 84.3]; %same
%clims = [81.3 84.3]; %3dB
```

```

aspect = [num2str(Fo),'Hz Source Theta 90° Phi 90° - Normalized Hybrid'];
figaspect = 'hybrid.90-90';
%aspect = ['3dB Beamwidth - ',num2str(Fo),'Hz Source Theta 90° Phi 90° -
Hybrid'];
%figaspect = 'hybrid.3dB.90-90';
%Omni
clims2 = [77.4 84.3]; %same
%clims2 = [77.4 80.4]; %3dB
aspect2 = [num2str(Fo),'Hz Source Theta 90° Phi 90° - Normalized Omni'];
figaspect2 = 'omni.90-90';
%aspect2 = ['3dB Beamwidth - ',num2str(Fo),'Hz Source Theta 90° Phi 90° -
Omni'];
%figaspect2 = 'omni.3dB.90-90';
amplitude = '1';
sensor = '323 & 324';
fid = fopen('E:\6 Nov 2009\VPV_12chan_N-65536_pass-30_Fs-
25.6k_1kHz_90theta_90phi_beam_set03.dat','r');
frewind(fid) %top of file

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%calculated parameters and constants
k = linspace(0,N-1,N); %baseline FFT bin numbers
d = 0.172; %array element spacing meters - measured 172mm ± 2mm
c = 340; %speed of sound meters/sec @ 20°C sea level
Fplot = floor(Fo*binN/Fs) + 1; %k bin for freq Fo
HBW = Fo/300; %+- freq to offset signal from Fo - broad for live signals
halfbandwidth = floor(HBW*binN/Fs) + 1; %+- number of bins to offset signal
maxchan = numsens * numel; %number of element channels, 6-8 are dummys
here
x = 1; %X axis designator
y = 2; %Y axis designator
z = 3; %Z axis designator

%Vector Sensor Angles relative to Array Apparatus - radians
%323 sensor
thetab1 = 0; %323 blu1 0°
phib1 = pi/2; %323 blu1 90°
thetag1 = pi/2; %323 grn1 90°
phig1 = 0; %323 grn1 0°
thetar1 = pi/2; %323 red1 90°
phir1 = pi/2; %323 red1 90°
%324 sensor
thetab3 = 0; %324 blu3 0°
phib3 = pi/2; %324 blu3 90°

```

```

thetag3 = pi/2; %324 grn3 90°
phig3 = 0; %324 grn3 0°
thetar3 = pi/2; %324 red3 90°
phir3 = pi/2; %324 red3 90°

%Sensor Element Unit Vectors
ue(prs1,:) = [0 0 0]; %323 prs1 dummy [0 0 0] - change in udot to [omni]
ue(blu1,:) = [sin(thetab1)*cos(phib1) sin(thetab1)*sin(phib1) cos(thetab1)]; %323
blu1 [0 0 1]
ue(grn1,:) = [sin(thetag1)*cos(phig1) sin(thetag1)*sin(phig1) cos(thetag1)]; %323
grn1 [1 0 0]
ue(red1,:) = [sin(thetar1)*cos(phir1) sin(thetar1)*sin(phir1) cos(thetar1)]; %323
red1 [0 1 0]
ue(prs2,:) = [0 0 0]; %aco prs2 dummy [0 0 0] - change in udot to [omni]
ue(blu2,:) = [0 0 0]; %aco no vel dummy
ue(grn2,:) = [0 0 0]; %aco no vel dummy
ue(red2,:) = [0 0 0]; %aco no vel dummy
ue(prs3,:) = [0 0 0]; %324 prs3 dummy [0 0 0] - change in udot to [omni]
ue(blu3,:) = [sin(thetab3)*cos(phib3) sin(thetab3)*sin(phib3) cos(thetab3)]; %324
blu3 [0 0 1]
ue(grn3,:) = [sin(thetag3)*cos(phig3) sin(thetag3)*sin(phig3) cos(thetag3)]; %324
grn3 [1 0 0]
ue(red3,:) = [sin(thetar3)*cos(phir3) sin(thetar3)*sin(phir3) cos(thetar3)]; %324
red3 [0 1 0]

%Steered Unit Vectors
thetas = 0:deg2rad(thetainc):pi; %polar steer angle array - radians
phis = -pi:deg2rad(phiinc):pi; %azimuthal steer angle array - radians
m = length(thetas); %total number of angles in theta direction
n = length(phis); %total number of angles in psi direction
p = length(k); %vector length in k direction
u = sin(thetas)' * cos(phis); %steered direction cosine x - m x n matrix
v = sin(thetas)' * sin(phis); %steered direction cosine y - m x n matrix
w = cos(thetas)' * ones(1,n); %steered direction cosine z - m x n matrix

%element unit vector & steer angle dot product array
%calculate one udot per element, same for all sensors,
%use first four channels of ue matrix
%vectorize dot products for faster multiplication in beamformer
udot = ue(prs1:red1,x)*u(:)' + ue(prs1:red1,y)*v(:)' + ue(prs1:red1,z)*w(:)'; %1-4 x
m*n
%omni-directional pressure sensors - vector is "1" in all orientations and aspects
udot(1,:) = ones(1,m*n); %1 x m*n

%Sensor_Transfer_Functions single .mat file with all transfer functions

```

```

%each velocity sensor transfer functions already multiplied by
%pressure transfer function
%orientation w/ src normal to yz plane used for computing pressure transfer
function
load 'Sensor_Transfer_Functions_old'; %1 x N length vectors

%+- bins around design freq to truncate - processing bandwidth, ignore
remainder
kshort = k(:,Fplot-halfbandwidth:Fplot+halfbandwidth); %get relevant k values
k = kshort; %transfer relevant k values to new k vector
clear kshort
p = length(k); %new vector length in k direction

%pre calculate phase weight constants matrix
phase_constant = 1i*2*pi*d*Fs/(c*binN); %constant term for alpha phase
weighting
theta_k = cos(thetas).' * k; %polar angle vs freq bin matrix for alpha phase
weighting
alpha_matrix = phase_constant * theta_k; %constant weighting matrix, multiply
by sensor

figure('Position',[300 100 800 600]) %Set Image size

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Start BeamFormer loop
fignum=1;

while fignum < numpass+1
    Xc = zeros(maxchan,N);

    %%load/read time domain signals from binary file%%
    %get time series datapoints binN x maxchan

    % For 2^16 binN files
    bindata = fread(fid, [binN,maxchan] , 'float64', 'ieee-be');

    %hanning window to reduce sidelobes of signal
    W = hann(binN) * ones(1,maxchan);
    xwin = bindata .* W;
    clear bindata W

    %convert binary signal data to frequency domain
    Xw = fft(xwin); %by columns binN x maxchan

```

```

%truncate length of signal to match transfer function length N x maxchan
X = Xw(1:N,:);
X = X.'; %change to row data maxchan x N, non conjugate transpose
clear xwin Xw %save memory

%Correct Signals to Aco mic
Xc(prs1,:) = X(prs1,:) .* Hp1;
Xc(blu1,:) = X(blu1,:) .* Hb1; %Hb1=Hp1*Hb1 - pre multiply vel and prs Xfer
func
Xc(grn1,:) = X(grn1,:) .* Hg1; %Hg1=Hp1*Hg1 - pre multiply vel and prs Xfer
func
Xc(red1,:) = X(red1,:) .* Hr1; %Hr1=Hp1*Hr1 - pre multiply vel and prs Xfer
func
Xc(prs2,:) = X(prs2,:);
Xc(prs3,:) = X(prs3,:) .* Hp3;
Xc(blu3,:) = X(blu3,:) .* Hb3; %Hb3=Hp3*Hb3 - pre multiply vel and prs Xfer
func
Xc(grn3,:) = X(grn3,:) .* Hg3; %Hg3=Hp3*Hg3 - pre multiply vel and prs Xfer
func
Xc(red3,:) = X(red3,:) .* Hr3; %Hr3=Hp3*Hr3 - pre multiply vel and prs Xfer
func

%truncate data to relevant bandwidth around Fo, ignore remainder
Xcshort = Xc(:,Fplot-halfbandwidth:Fplot+halfbandwidth);
Xc = Xcshort;
clear Xcshort %save memory
Xc = conj(Xc);

%%initialize beamformer%%
%%S is sum of all elements and sensors - the hybrid beamformer%%
%%S2 is sum of omni elements and sensors - the omni beamformer%%
S = zeros(m*n,size(Xc,2)); %final hybrid beam former output - m*n x p array
S2 = zeros(m*n,size(Xc,2)); %final omni beam former output - m*n x p array

%Beamformer output

%%323 Sensor%%
%alpha - weighting same for all elements of a single sensor
%for sensor 323 m x p same length as signal
alpha = exp(((ceil(prs1/4)-2))*alpha_matrix);
alphacalc = conj(alpha((1:m)' * ones(1,n),:)); %copy alpha matrix to m*n x p
for q = prs1:red1; %1-4
    %pull element matrix from udot array
    %Theta m * Phi n vector

```

```

        udotc = squeeze(udot((mod(q+3,4)+1),:,:)); %m*n x 1
        Xcalc = udotc(:) * Xc(q,:); %m*n x p
        %Theta m x FFT o matrix
        %sumcalc - individual element output for each angle combo m*n x p
matrix
        %indexing alpha matrix to m*n x p faster than using "repmat" with same
effect
        sumcalc = (Xcalc) .* (alphacalc); %conjugate of alpha taken due to
transpose
        %S - total beamformer output for each angle combo m*n x p matrix
        S = S + sumcalc;
        if q == prs1
            S2 = S2 + sumcalc;
        end
    end
end

%%ACO Sensor%%
%center element, alpha = 1, remove to speed processing
q = prs2; %5
    %pull element matrix from udot array
    %Theta m * Phi n vector
    udotc = squeeze(udot((mod(q+3,4)+1),:,:)); %m*n x 1
    sumcalc = udotc(:) * Xc(q,:);
    %S - total beamformer output for each angle combo m*n x p matrix
    S = S + sumcalc;
    S2 = S2 + sumcalc;

%%324 Sensor%%
%alpha - weighting same for all elements of a single sensor
%for sensor 324 mxp same length as signal
alpha = exp(((ceil(prs3/4)-2))*alpha_matrix);
alphacalc = conj(alpha((1:m)' * ones(1,n),:)); %copy alpha matrix to m*n x p
for q = prs3:red3; %9-12
    %pull element matrix from udot array
    %Theta m * Phi n vector
    udotc = squeeze(udot((mod(q+3,4)+1),:,:)); %m*n x 1
    Xcalc = udotc(:) * Xc(q,:); %m*n x p
    %pull element matrix from phase weights array
    %Theta m x FFT o matrix
    %sumcalc - individual element output for each angle combo m*n x p
matrix
    %indexing alpha matrix to m*n x p faster than using "repmat" with same
effect
    sumcalc = (Xcalc) .* (alphacalc); %conjugate of alpha taken due to
transpose

```

```

    %S - total beamformer output for each angle combo m*n x p matrix
    S = S + sumcalc;
    if q == prs3
        S2 = S2 + sumcalc;
    end
end
end

%%Beamformer output%%

S = sum(abs(S),2); %sum along k "amplitude", collapse to single value in k
direction
S = reshape(S, [m n]); %mxn thetaxphi 2D matrix
SdB = 20*log10(abs(S)); %convert to dB
S = S/max(max(S)); %normalize
S2 = sum(abs(S2),2); %sum along k "amplitude", collapse to single value in k
direction
S2 = reshape(S2, [m n]); %mxn thetaxphi 2D matrix
S2dB = 20*log10(abs(S2)); %convert to dB
S2 = S2/max(max(S2)); %normalize

%Find 3dB beamwidths of main lobe
%maximum response
[SdBmax1,l1] = max(max(SdB,[],1));
[SdBmax2,l2] = max(max(S2dB,[],1));

%convert to cartesian coordinates
%r = 1;
xdir = S .* u; %x coord normal to yz plane
ydir = S .* v; %y coord normal to xz plane
zdir = S .* w; %z coord normal to xy plane
xdir2 = S2 .* u; %x coord normal to yz plane
ydir2 = S2 .* v; %y coord normal to xz plane
zdir2 = S2 .* w; %z coord normal to xy plane

%Overhead Theta by Phi amplitude plot
imagesc(rad2deg(phis),rad2deg(thetas),SdB,clims);
title(aspect)
axis([-180 180 0 180])
xlabel('\phi (deg)')
ylabel('\theta (deg)')
shading interp
moviesc(fignum) = getframe;
saveas(gcf,
['BeamFormer.XV.imagesc.',num2str(Fo),'Hz.',figaspect,'.',num2str(fignum),'.fig'],
'fig');

```

```

imagesc(rad2deg(phis),rad2deg(thetas),S2dB,clims2);
title(aspect2)
axis([-180 180 0 180])
xlabel('\phi (deg)')
ylabel('\theta (deg)')
shading interp
moviesc2(fignum) = getframe;
saveas(gcf,
['BeamFormer.XV.imagesc.',num2str(Fo),'Hz.',figaspect2,'.',num2str(fignum),'.fig']
, 'fig');

%3D rendering of composite array beam pattern cartesian coordinates
surf(ydir,xdir,zdir,SdB,'EdgeColor','none'); %nxm 3D plot at freq of interest
title(aspect)
xlabel('x axis')
ylabel('y axis')
zlabel('z axis')
shading interp
axis([-1.1 1.1 -1.1 1.1 -1.1 1.1])
view(180,0)
movie3d(fignum) = getframe;
saveas(gcf,
['BeamFormer.XV.3d.',num2str(Fo),'Hz.',figaspect,'.',num2str(fignum),'.fig'], 'fig');

surf(ydir2,xdir2,zdir2,S2dB,'EdgeColor','none'); %nxm 3D plot at freq of
interest
title(aspect2)
xlabel('x axis')
ylabel('y axis')
zlabel('z axis')
shading interp
axis([-1.1 1.1 -1.1 1.1 -1.1 1.1])
view(180,0)
movie3d2(fignum) = getframe;
saveas(gcf,
['BeamFormer.XV.3d.',num2str(Fo),'Hz.',figaspect2,'.',num2str(fignum),'.fig'], 'fig');

fignum = fignum + 1;
end
%End BeamFormer loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Save movie

```

```

save(['Beampattern_Movie.',num2str(Fo),'.',figaspect,'.combo.mat'], 'moviesc',
'movie3d', 'moviesc2', 'movie3d2', 'SdBmax1', 'SdBmax2', 'clims', 'clims2')
% Play Movie code
%imagesc images
%figure('Position',[300 100 800 600]) %Set Image size
%axis([-180 180 0 180])
%xlabel('\phi (deg)')
%ylabel('\theta (deg)')
%movie(moviesc,1,1.3)
%movie(moviesc2,1,1.3)

%3D images
%figure('Position',[300 100 800 600]) %Set Image size
%axis([-1.1 1.1 -1.1 1.1 -1.1 1.1])
%xlabel('x axis')
%ylabel('y axis')
%zlabel('z axis')
%view(180,0)
%movie(movie3d,1,2.56)
%movie(movie3d2,1,2.56)

movie2avi(moviesc, ['moviesc.',num2str(Fo),'.',figaspect,'.avi'],'FPS',2.56)
movie2avi(movie3d, ['movie3d.',num2str(Fo),'.',figaspect,'.avi'],'FPS',2.56)
movie2avi(moviesc2, ['moviesc2.',num2str(Fo),'.',figaspect2,'.avi'],'FPS',2.56)
movie2avi(movie3d2, ['movie3d2.',num2str(Fo),'.',figaspect2,'.avi'],'FPS',2.56)

status = fclose('all'); % close all files
close %close figure window

```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

- ACO Pacific Inc. (n.d.). *ACOtron™ Preamp Specifications, 4012 Family*. [Brochure]. Retrieved from <http://www.acopacific.com/acotrndt.html#4012> on 1 Nov 2009.
- Calculated Industries. (n.d.). *Laser Distance Meter 3350 Specification Sheet*. [Brochure]. Retrieved from http://www.calculated.com/artwork/3350-SPEC-4C-03_09.pdf on 1 Sep 2009.
- Cray, B. A. & Nuttall, A. H. "Directivity factors for linear arrays of velocity sensors," J. Acoust. Soc. Am. 110, pp. 324–331, 2001.
- Flightaware. (n.d.). [Search term: Monterey Airport]. Retrieved from <http://flightaware.com/live/airport/KMRY> on 12 Nov 2008.
- Hawkes, M. & Nehorai, A. "Wideband source localization using a distributed acoustic vector-sensor array," IEEE Trans. Sig. Proc., pp. 1479–1491, 2003.
- Hawkes, M. & Nehorai, A. "Acoustic vector-sensor beamforming and capon direction estimation," IEEE Trans. Sig. Proc. 46, pp. 2291–2304, 1998.
- Kinsler, L. E., Frey, A. R, Coppens, A. B., & Sanders, J. V. (2000). *Fundamentals of Acoustics*. (5th ed.) New York: John Wiley and Sons.
- Lurton, X. (2002). *An Introduction to Underwater Acoustics – Principles and Applications*. Chichester, UK: Praxis Publishing.
- Mann, J. A., Tichy, J., & Romano, A. J., "Instantaneous and time-averaged energy transfer in acoustic fields," J. coust. Soc. Am. 82, pp. 17–29, 1987.
- Microflown Holland. (n.d.). *Ultimate Sound Probe Datasheet*. [Brochure]. Retrieved from http://www.microflown.com/data/manuals_datasheets/probes/Datasheet_USP_v1.0.pdf on 15 May 2009.
- Microflown Holland. (n.d.). *Ultimate Sound Probe Manual*. [Brochure]. Retrieved from http://www.microflown.com/data/manuals_datasheets/probes/Manual_USP%20regular_V1.0.pdf on 15 May 2009.

- Microflown Holland. (n.d.). *Four Channel Signal Conditioner Datasheet*. [Brochure]. Retrieved from http://www.microflown.com/data/manuals_datasheets/signal_conditioners/Datasheet_4channel_signal_conditioner_v1.0.pdf on 15 May 2009.
- Microflown Holland. (n.d.). *Four Channel Signal Conditioner Manual*. [Brochure]. Retrieved from http://www.microflown.com/data/manuals_datasheets/signal_conditioners/Technical_specs_channel_signal_conditioner.pdf on 15 May 2009.
- National Instruments. (n.d.). *NI cDAQ-9172 Technical Specifications*. [Brochure]. Retrieved from http://www.ni.com/pdf/products/us/ni_compactdaq.pdf on 15 May 2009.
- National Instruments. (n.d.). *NI cDAQ-9172 User Guide*. [Brochure]. Retrieved from <http://www.ni.com/pdf/manuals/371747f.pdf> on 15 May 2009.
- National Instruments. (n.d.). *NI USB-9234 Technical Specifications*. [Brochure]. Retrieved from http://www.ni.com/pdf/products/us/cat_sv923x.pdf on 15 May 2009.
- National Instruments. (n.d.). *NI USB-9234 User Guide*. [Brochure]. Retrieved from <http://www.ni.com/pdf/manuals/374238c.pdf> on 15 May 2009.
- Nehorai, A., "Acoustic vector-sensor array processing," *IEEE Trans. Sig. Proc.* 42, pp. 2481–2491, 1994.
- Peebles, P. Z. (2001). *Probability, Random Variables and Random Signal Principles*. (4th ed.) New York: McGraw-Hill.
- Schmidt, R., "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas and Propagation* 34, pp. 276–280, 1986.
- Smith, K. B. (2009), personal notes.
- Smith, S. W. (2003). *Digital Signal Processing, A Practical Guide for Engineers and Scientists*. Oxford: Newnes.
- Urick, R. J. (1983). *Principles of Underwater Sound*. (3rd ed.) Los Altos: Peninsula Publishing.
- Vifa Denmark. (n.d.). *Vifa 5" Woofer - model P13WH-00-08 Datasheet*. [Brochure]. Retrieved from <http://www.parts-express.com/pdf/297-302.pdf> on 1 Sep 2009.

Wind, J. W., Tijs, E., & de Bree, H.-E., "Source localization using acoustic vector sensors: A MUSIC approach," Noise and Vibration: Emerging Methods Conference Proceedings, 5–8 April 2009 [in press].

Ziomek, L. J. (1994). *Fundamentals of Acoustic Field Theory and Space-Time Signal Processing*. Boca Raton: CRC Press, p. 413.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California
3. Hans-Elias de Bree
Microflow Holland
Arnhem, Netherlands
4. Daphne Kapolka
Naval Postgraduate School
Monterey, California
5. Kevin Smith
Naval Postgraduate School
Monterey, California
6. Robert Barton
Naval Undersea Warfare Center
New Port, Rhode Island
7. Ellen Livingston
Office of Naval Research
Arlington, Virginia
8. Mike Traweek
Office of Naval Research
Arlington, Virginia
9. Jeff Caulk
Naval Postgraduate School
Monterey, California